

BÁO CÁO ĐỒ ÁN

NHẬP MÔN KHOA HỌC DỮ LIỆU

20120040 Nguyễn Quang Gia Bảo

20120136 Huỳnh Tuấn Nam

20120158 Trần Hoàng Anh Phi

01. Thu thập dữ liệu
02. Khám phá dữ liệu
03. Đặt các câu hỏi có ý nghĩa
cần trả lời
04. Mô hình hóa dữ liệu
05. Đánh giá mô hình

THU THẬP DỮ LIỆU

Dữ liệu cho đồ án được thu thập từ website: tiki.vn

Đây là một sàn thương mại điện tử khá lớn ở Việt Nam, chuyên cung cấp rất nhiều mặt hàng đa ngành cho người tiêu dùng.

License

Theo như nội dung api mà tiki cung cấp cho nhà phát triển thì chúng ta được phép sử dụng api để thu thập dữ liệu.

THU THẬP DỮ LIỆU

Vì web có rất nhiều mặt hàng nên nhóm chỉ chọn danh mục **Sách tiếng Việt** để thu thập và thực hiện đồ án

Đường dẫn đến danh mục Sách tiếng Việt:
<https://tiki.vn/sach-truyen-tieng-viet/c316>

THU THẬP DỮ LIỆU

Bước 1:

Lấy các danh mục con của danh mục sách tiếng Việt.

Bước 2:

Nếu các danh mục có danh mục con thì thay thế bằng các danh mục con này.

Bước 3:

Lấy id của các cuốn sách dựa vào các danh mục đã thu thập.

Bước 4:

Crawl thông tin của từng cuốn sách dựa vào id vừa thu thập.

THU THẬP DỮ LIỆU

Bước 1: Lấy các danh mục con của danh mục Sách tiếng Việt.

- Viết hàm `generate_url(category, page, urlKey)` để trả về đường dẫn đến danh mục dựa vào các tham số:

```
DEF GENERATE_URL(CATEGORY, PAGE, URLKEY):  
    RETURN F'HTTPS://TIKI.VN/API/PERSONALISH/V1/BLOCKS/LISTINGS?  
LIMIT=100&AGGREGATIONS=2&SORT=TOP_SELLER&CATEGORY={CATEGORY}&PAGE={PAGE}&URLKEY={URLKEY}'
```

- Tạo request lên đường dẫn của danh mục Sách tiếng Việt, sau đó lọc các giá trị của key "filters" -> "display_name", sau đó lấy danh sách các danh mục.

THU THẬP DỮ LIỆU

Bước 2: Lấy các danh mục con của các danh mục đã thu thập.

- Lặp lần lượt qua các danh mục đã lấy và thực hiện các bước tương tự như bước 1 để lấy thêm các danh mục con (nếu có)

THU THẬP DỮ LIỆU

Bước 3: Lấy id của các cuốn sách dựa vào các danh mục đã thu thập.

- Viết hàm `get_books_id(cat)` nhận tham số đầu vào là `cat` - tên danh mục, từ tên danh mục này generate ra các tham số cho đường dẫn api. Request lên đường dẫn và lấy file json. Từ file json này lấy ra các id thuộc danh mục đó.
- Lặp lần lượt qua các category để lấy toàn bộ id.
- Lưu id vào file csv để phục vụ cho việc lấy dữ liệu

THU THẬP DỮ LIỆU

Bước 4: Crawl thông tin của từng cuốn sách dựa vào id đã thu thập

- Viết hàm `get_book_info(raw_book)` nhận tham số `raw_book` - file json lấy được từ mỗi id.
- Xác định các thuộc tính cần lấy của một cuốn sách.
- Với các thuộc tính đặc biệt (tên tác giả, thông số kỹ thuật) ta xử lý riêng.
- Vì số lượng id khá lớn, dễ có lỗi trong quá trình thu thập nên phải viết hàm `split_books_id(books_id: list, i, n)` để chia id ra làm các phần và xử lý lần lượt.

THU THẬP DỮ LIỆU

- Viết hàm `get_all_books(book_df, start, end)` để lấy thông tin của các cuốn sách từ vị trí `start` tới `end`, trả về danh sách các cuốn sách trong `dataframe book_df`.
- Ở đây em chia id thành 5 phần và lần lượt crawl sau đó lưu vào các file csv theo tên là `part_1.csv, part_2.csv,...`
- Cuối cùng, concat các file lại và lưu thành file `data.csv`

KHÁM PHÁ VÀ TIỀN TIỀN XỬ LÝ DỮ LIỆU

- Cột 'Unnamed:0' có giá trị trùng với số thứ tự dòng, trường hợp này có 2 hướng giải quyết:
 - Xóa hẳn cột 'Unnamed:0' (quyết định của nhóm)
 - Đặt nó thành 1 index mới

KHÁM PHÁ VÀ TIỀN XỬ LÝ DỮ LIỆU

```
df = pd.read_csv('full_data/data.csv')  
print(df.shape)  
display(df.head())
```

✓ 0.9s

```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_21316\906639567.py:1: DtypeWarning: Columns
```

```
(22,27,29,30,31,32,33,34,35,36,37,38,39,40,41,43,45,46,47,48,49,50,51,52,53,54,56,57,58,59,60,61,62,63,64,65,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83) have mixed
```

```
df = pd.read_csv('full_data/data.csv')
```

```
(96647, 84)
```

- Bộ data này có 96647 dòng và có 84 cột.
- còn (22, 27, 29, ...) là các cột bị mixed type.

KHÁM PHÁ DỮ LIỆU

Vì dữ liệu lấy về từ API là dữ liệu thô nên ta cần phải xử lý để có thể sử dụng được. Đầu tiên, ta sẽ xem có các hàng trùng nhau hay không.

```
df.duplicated().sum()
```

8294

Có 8294 dòng trùng, ta sẽ xử lý bằng cách drop các dòng này đi.

KHÁM PHÁ VÀ TIỀN & TIỀN XỬ LÝ DỮ LIỆU

Kiểm tra các cột dữ liệu, có rất nhiều cột **lạc loài**.

```
Index(['id', 'master_id', 'sku', 'name', 'short_url', 'book_cover', 'price',  
       'original_price', 'discount_rate', 'rating_average', 'review_count',  
       'inventory_type', 'productset_group_name', 'day_ago_created',  
       'categories', 'all_time_quantity_sold', 'authors', 'publisher_vn',  
       'publication_date', 'dimensions', 'manufacturer', 'number_of_page',  
       'dich_gia', 'edition', 'brand', 'brand_country', 'item_model_number',  
       'origin', 'product_weight', 'battery_capacity', 'camera', 'chong_nuoc',  
       'loai_day', 'san_pham', 'storage', 'battery_life', 'charge_time',  
       'do_chiu_nuoc', 'included_accessories', 'material', 'luu_y',  
       'minimum_inbound_policy_days', 'expiry_time', 'shelf_life_days',  
       'chat_lieu', 'audio_power_output', 'audio_technology'],
```

```
       'image_processing_technology', 'network_name', 'network_internet',  
       'network_wifi', 'remote_thong_minh', 'resolution', 'screen_mirroring',  
       'screen_size', 'size_without_stand_table_top',  
       'size_with_stand_table_top', 'tivi_type', 'ung_dung', 'usb',  
       'weight_without_stand', 'weight_with_stand', 'size', 'loai_pin',  
       'thoi_gian_su_dung', 'capacity', 'huong_dan_su_dung', 'kich_thuoc',  
       'tai_trong', 'device_brand', 'do_nhay', 'tan_so', 'huong_dan_bao_quan',  
       'dac_diem_noi_bat', 'dieu_kien_su_dung', 'dia_chi_su_dung',  
       'bluetooth'],
```

KHÁM PHÁ VÀ TIỀN TIỀN XỬ LÝ DỮ LIỆU

Các cột mà ta sử dụng: 'id', 'master_id', 'sku', 'name', 'short_url',
'book_cover', 'price', 'original_price', 'discount_rate', 'rating_average',
'review_count', 'inventory_type', 'productset_group_name',
'day_ago_created', 'categories', 'all_time_quantity_sold', 'authors',
'publisher_vn', 'publication_date', 'dimensions', 'manufacturer',
'number_of_page', 'dich_gia', 'edition', 'luu_y'

Trừ các dòng mà ta cần, còn lại tất cả các cột và dòng khác ta xóa tất cả.

KHÁM PHÁ VÀ TIỀN TIỀN XỬ LÝ DỮ LIỆU

Ta giữ lại các dòng "categories" hợp lý (đó là các thể loại sách).

```
valid_categories = pd.read_csv('id_data/categories_id.csv')['categories']
valid_categories = '|'.join(list(valid_categories))
df = df[df['categories'].str.contains(valid_categories, na=False)]
```

Kết quả: Ta chỉ còn 48290 dòng và 25 cột.

KHÁM PHÁ VÀ TIỀN TIỀN XỬ LÝ DỮ LIỆU

Ta giữ lại các dòng "categories" hợp lý (đó là các thể loại sách).

```
valid_categories = pd.read_csv('id_data/categories_id.csv')['categories']
valid_categories = '|'.join(list(valid_categories))
df = df[df['categories'].str.contains(valid_categories, na=False)]
```

Kết quả: Ta chỉ còn 48290 dòng và 25 cột.

KHÁM PHÁ VÀ TIỀN XỬ LÝ DỮ LIỆU

Tìm hiểu sự liên quan giữa cột id và master_id

```
df[~(df["id"] == df["master_id"])]
```

148]

```
...   id  master_id  sku  name  short_url  book_cover  price  original_price  discount_rate  rating_average  ...  all_time_quantity_sold
```

0 rows × 25 columns

Loại bỏ cột master_id

KHÁM PHÁ DỮ LIỆU

Các cột có ý nghĩa gì?

id	id của quyển sách
sku	mã hàng hoá
name	tên của quyển sách
short_url	đường link đến sách
book_cover	bìa sách
price	giá sách
original_price	giá gốc của sách
discount_rate	% giảm giá của sách
rating_average	mức đánh giá trung bình
review_count	số lượt đánh giá
inventory_type	tình trạng kho
productset_group	tên nhóm sản phẩm
group_name	
day_ago_created	số ngày từ khi đăng bán
categories	loại sách
all_time_quantity	số lượng sách đã bán
total_sold	
authors	tác giả
publisher_vn	công ty phát hành
publication_date	ngày xuất bản
dimensions	kích thước

KHÁM PHÁ DỮ LIỆU

Các cột có ý nghĩa gì?

manufacturer	nhà xuất bản
number_of_page	số trang
dich_gia	tên dịch giả
edition	phiên bản của sách
luu_y	những lưu ý thay đổi trên sách

KHÁM PHÁ VÀ TIỀN TIỀN XỬ LÝ DỮ LIỆU

Các cột thuộc kiểu số (numeric):

- 'id', 'sku', 'price', 'original_price', 'discount_rate', 'rating_average', 'review_count', 'day_ago_created', 'all_time_quantity_sold'

Cột **number_of_page** biểu diễn lại số trang sách. Nhưng nó không có ở trong các cột numeric.

```
set(df['number_of_page'].apply(lambda x: type(x)))
```

Kiểm tra dữ liệu lại các kiểu dữ liệu bên trong cột thì thấy được rằng: cột này có hai kiểu dữ liệu **float, str**

KHÁM PHÁ VÀ TIỀN XỬ LÝ DỮ LIỆU

Tại sao lại có giá trị str

```
df[df['number_of_page'].apply(lambda x: isinstance(x, str))]['number_of_page'].unique()
```

```
array(['336', '178', '216', ..., '10512', '1058', '879'], dtype=object)
```

Kiểm tra lại trong các giá trị này, có giá trị nào mang chữ cái không

```
df[df['number_of_page'].str.contains(r'[a-zA-Z]', na=False)]['number_of_page'].unique()
```

```
array(['Cuốn', '80x2', 'mềm'], dtype=object)
```

KHÁM PHÁ VÀ TIỀN XỬ LÝ DỮ LIỆU

Cột "publication_date" mang kiểu dữ liệu object.

```
df.publication_date = pd.to_datetime(df.publication_date, format = "%Y-%m-%d  
%H:%M:%S", errors = "coerce")
```

Sử dụng errors = "coerce" để những giá trị lỗi được đưa về NaT
Và có 29534 giá trị NaT. Việc này xảy ra là do một số cột bị lỗi hoặc có giá trị là
Nan.

KHÁM PHÁ VÀ TIỀN XỬ LÝ DỮ LIỆU

Các mô tả về cột số

	id	sku	price	original_price	discount_rate	rating_average	review_count	day_ago_created	all_time_quantity_sold	number_of_page
missing_ratio	0.0	0.000000e+00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
min	148562.0	1.000110e+12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
max	207990688.0	9.999995e+12	6000000.0	6950000.0	91.0	5.0	6306.0	2689.0	32323.0	9.786049e+12

Các cột số tất cả đều không có missing.

KHÁM PHÁ VÀ TIỀN & TIỀN XỬ LÝ DỮ LIỆU

Mô tả các cột còn lại.

	authors	book_cover	categories	dich_gia	dimensions	edition	inventory_type	luu_y	manufacturer	name	productset_group_name	publication_date	publisher_vn	short_url	
missing_ratio	48.804125	35.358555	0.0	85.686774	82.744197	96.931105	0.0	99.995858	0.550827	0.0	0.0	61.158394	0.035203	0.0	
num_diff_vals	9094	17	100	3110	1051	364	3	1	168	44424	[1111	101	10896	852	48288
diff_vals	[Trần Đặng Đặng Khoa, Trần Hồng Ngọc, Lê Quang...]	[Bìa mềm, Bìa Da, Bìa cứng, Bìa gập, Bìa rời, ...]	[Du ký, Light novel, Phê Bình - Lý Luận Văn Họ...]	[Khánh Vân, Phan Quang, Nguyễn Thị Bách Tuyết...]	[13 x 20, 14,5 x 20,5 cm, 13,5 x 20,5 cm, 15 x...]	[bìa mềm, BÌA MỀM, Tiếng Việt, Tặng kèm bookma...]	[instock, backorder, preorder]	[Sách không còn kèm CD mà thay bằng ứng dụng t...]	[NXB Trẻ, Nhà Xuất Bản Lao Động, Nhà Xuất Bản ...]	Nhật Ký Sau Văn Đàm Trên Yên Xe Cà Tân...	[Nhà Sách Tiki/Sách tiếng Việt/Sách văn học/Du...]	[2022-11-10 00:00:00, 2022- 11-11 17:44:46, 202...	[NXB Trẻ, Chibooks, Văn Lang, Edibooks, NXB Ph...]	[https://tiki.vn/product- p204317934.html?spid=...]	

KHÁM PHÁ VÀ TIỀN & TIỀN XỬ LÝ DỮ LIỆU

Trong quá trình xét các diff_vals trong các cột. Thì cột "dimension" thật sự có vấn đề.

dimensions	
0	13 x 20
1	NaN
2	14,5 x 20,5 cm

Vừa có cm vừa không có cm ???

Giải quyết: Ta sẽ đưa dữ liệu dưới dạng string thành list.

[13, 20]

⋮

[14,5, 20,5]

KHÁM PHÁ VÀ TIỀN XỬ LÝ DỮ LIỆU

Thay đổi giá trị cột dimension

```
dimen_df["regex_dimen"] = df.dimensions.apply(lambda x: re.findall(r"[0-9]+[.]?[0-9]+|/", str(x)))
```

- dimen_df là một dataframe mới để ta thực hiện các thay đổi trước khi thay đổi giá trị gốc.

Tuy nhiên, có 16 dòng xảy ra vấn đề

3175	<p><span style="font-size: 11pt; font-family: ...	[11, 13.5, 20, 14720, 10, 11, 14, 2, 0, 15, 16, ...]
4012	<p><span style="font-size: 11pt; font-family: ...	[11, 13.5, 20, 14720, 10, 11, 14, 2, 0, 15, 16, ...]
6485	<p>20.5 x 14.5 x 5.0 cm</p>	[20.5, 14.5, 5.0, /]
12396	<p class="MsoNormal">14...	[14, /, /, 20.5, /, /, /]

Ta xử lý bỏ đi "/". Tuy nhiên, dữ liệu vẫn không làm sạch được do có các số khác từ datafarme gây nhiễu.

KHÁM PHÁ VÀ TIỀN XỬ LÝ DỮ LIỆU

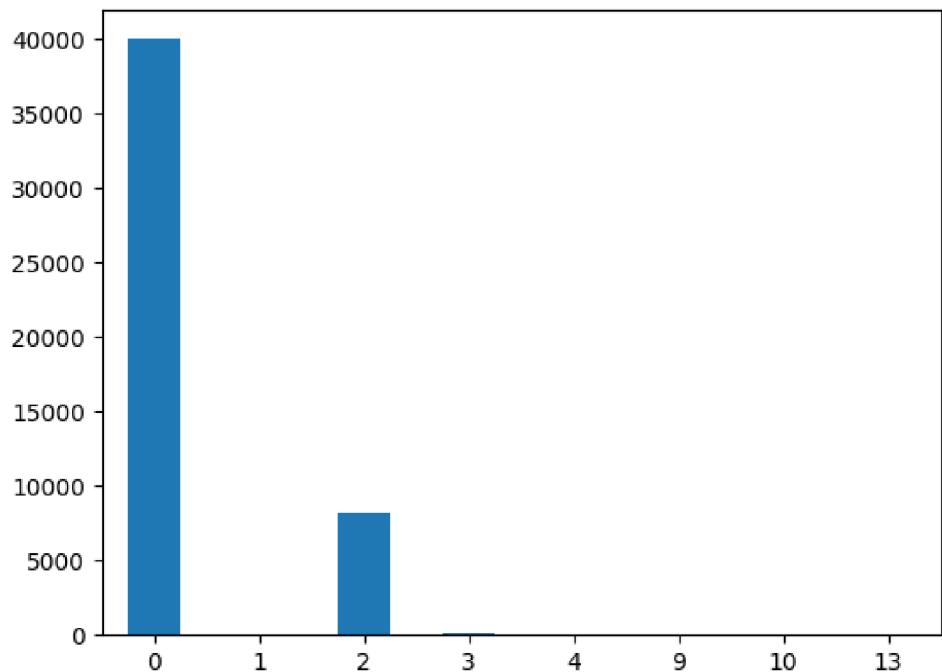
Giải quyết vấn đề dấu "/" hoặc các kí tự không phù hợp

Hàm `SplitNumber` sẽ giải quyết các vấn đề này

```
def SplitNumber(dimension):  
    regex_dimen = []  
    for d in dimension:  
        if d.isnumeric():  
            regex_dimen.append(d)  
        elif d.strip('0123456789') == '.' or d.strip('0123456789') == ',':  
            regex_dimen.append(d)  
    return regex_dimen
```

KHÁM PHÁ VÀ TIỀN & TIỀN XỬ LÝ DỮ LIỆU

Sau khi xử lí:



Giá trị 0 rất cao

Theo missing_ratio có hơn 83% giá trị
thiếu

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Trả lời câu hỏi thành công:

- Nếu trả lời được câu hỏi bạn sẽ hiểu tại sao trong bộ dữ liệu lại có nhiều sách có cùng tên? Chúng khác nhau điều gì ?

Khó khăn:

- Tên của những quyển sách lại được đặt khác nhau. Ảnh hưởng đến quá trình gom nhóm. Cần xử lý vấn đề này cho một bộ dữ liệu gần 40 nghìn dữ liệu.
- Không có các yếu tố để phân biệt tên của sách.

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Hướng giải quyết

- Thay vì xử lí một lúc 40 ngàn dữ liệu, ta sẽ sử dụng cột `categories` để phân loại thành từng loại sách, rồi tiếp tục phân loại theo tên.
- Ta sử dụng `Ratio Matching` - Độ tương thích của 2 string, để phán đoán series của cuốn truyện.

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Ta sẽ dùng **fuzz** thu viện **fuzzywuzzy** để thực hiện việc tìm kiếm tỉ lệ trùng khớp

Ta có 2 string:

```
a = "Xin Cảm ơn"  
b = "Xin Chào"
```

```
fuzz.ratio(a,b)
```

- Độ giống nhau là 56%
- Độ tương thích này được xét theo các yếu tố như:
 - Số lượng từ ở trong hai câu
 - Vị trí từ
 - Kiểu viết của chữ (hoa hoặc thường)

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Bước 1: Lọc Dataframe theo thể loại sách mà ta cần thống kê

```
def get_df_by_categories(categories):  
    categories_book_df = df[df.categories == categories]  
    return categories_book_df
```

Truyền vào categories hợp lệ với bộ dữ liệu của dataset, ta sẽ có được một dataframe với các thể loại sách mà ta muốn.

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Bước 2: Phân nhóm các quyển sách theo từng series, hoặc cùng tên với nhau

Giống như việc khi tra Google, nếu ta gõ tên một quyển sách, thì Google sẽ cho ta:

- Nếu cuốn sách có nhiều tập, thì Google sẽ cho ta các tập của cuốn truyện
- Nếu cuốn sách chỉ có 1 bản duy nhất, thì sẽ ra nhiều trang khác nhau.

Ta sử dụng việc này để ta thực hiện thuật toán này.

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Bước 2: Phân nhóm các quyển sách theo từng series, hoặc cùng tên với nhau

Cách thực hiện:

- Ta sẽ kiểm tra có 1 list để chứa các series và sẽ dùng list này để đi so sánh với tên của các cuốn sách khác.
- So sánh cuốn sách với từng series bên trong list. Nếu độ tương thích dưới 80% sẽ add tên đó vào series. Ngược lại ta sẽ thêm vào thì ta không thêm vào.
- Với mỗi series hoặc sách cùng tên trong list sẽ được đánh dấu 1 index, ta gọi hàm index của list để lấy index đó và gán cho dòng của cuốn sách đó.

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Bước 2: Phân nhóm các quyển sách theo từng series, hoặc cùng tên với nhau

```
def categorical_book_name(book_df):  
    u_names = list(map(lambda x: x.upper(), list(book_df.name)))  
    book_series = []  
    series = []  
    book_series.append(u_names[0])
```

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Bước 2: Phân nhóm các quyển sách theo từng series, hoặc cùng tên với nhau

```
for name in u_names:  
    ratio_series = process.extract(name, book_series, scorer = fuzz.token_sort_ratio)  
    best_ratio_series = ratio_series[0]  
    if best_ratio_series[-1] < 65: # Xét tỉ lệ hợp lệ cao hơn 80%  
        book_series.append(name)  
        series.append(book_series.index(name))  
    else:  
        series.append(book_series.index(best_ratio_series[0]))  
return series
```

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Bước 2: Phân nhóm các quyển sách theo từng series, hoặc cùng tên với nhau

Kết quả thu được: Khi ta chọn loại sách là **Light novel** thì kết quả ta thu được:

14,
15,
5,
16,
17,
0,
14,

series sẽ phân loại các cuốn sách theo một mã số phân loại

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Bước 3: Sau đó ta đưa list series vừa tạo ra vào Dataframe đang dùng để đánh dấu lại.

```
lightnovel_df.insert(len(lightnovel_df.columns), "Series", series)
```

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Ta truy cập vào từng series để có được sách mà ta muốn

```
lightnovel_df[lightnovel_df.Series == 0]
```

102	203957932	4966276852148	Mừng Đến Lớp Học Đề Cao Thực Lực - 4	https://tiki.vn/product-p203957932.html?spid=2...	Bìa mềm	95900	120000	20
121	195969669	9063209978122	Chào Mừng Đến Lớp Học Đề Cao Thực	https://tiki.vn/product-p195969669.html?spid=1...	Bìa mềm	95900	120000	20

TRẢ LỜI CÂU HỎI

Câu 1: Tại sao có những quyển sách cùng tên?

Đánh giá / Tự trả lời, đưa ra dự đoán cho câu hỏi

Tuy cùng là một cuốn sách, nhưng có nhiều shop khác nhau bán, có mức độ đánh giá trung bình, review của khách hàng, mức khuyến mãi, giá cả khác nhau... dẫn đến việc lựa chọn nên mua ở shop nào.

TRẢ LỜI CÂU HỎI

Câu 2: Lượt đánh giá trung bình, ảnh hưởng như thế nào?

- `rating-average` (Mức độ đánh giá trung bình), `original_price` (Giá gốc sản phẩm), `all_time_quanity_sold` (số lượng sản phẩm bán ra), `review_count` (số lượt bình luận về món hàng) có liên quan gì đến nhau ?

TRẢ LỜI CÂU HỎI

Câu 2: Lượt đánh giá trung bình, ảnh hưởng như thế nào?

Trả lời câu hỏi thành công:

- Sẽ hiểu được việc đánh giá của khách hàng sẽ có ảnh hưởng thế nào đến việc kinh doanh của shop.

TRẢ LỜI CÂU HỎI

Câu 2: Lượt đánh giá trung bình, ảnh hưởng như thế nào?

Các cột sử dụng:

- rating_average
- categories
- price
- original_price
- all_time_quantity_sold
- review_count

Các thư viện sử dụng:

- matplotlib.pyplot
- seaborn

TRẢ LỜI CÂU HỎI

Câu 2: Lượt đánh giá trung bình, ảnh hưởng như thế nào?

Bước 1: Tạo khoảng cho các lượt đánh giá trung bình

Các khoảng được chia

- (0,0) Ứng với 0 sao
- (0,1) Ứng với (0,1) sao
- (1,2) Ứng với [1,2) sao
- (2,3) Ứng với [2,3) sao
- (3,4) Ứng với [3,4) sao
- (4,5) Ứng với [4,5) sao
- (5,5) Ứng với 5 sao

Hàm xử lý

```
def rating_range(x):  
    if (x == 0.0):  
        return (0,0)  
    elif (x == 5.0):  
        return (5,5)  
    else:  
        return (math.floor(x), math.floor(x+1))
```

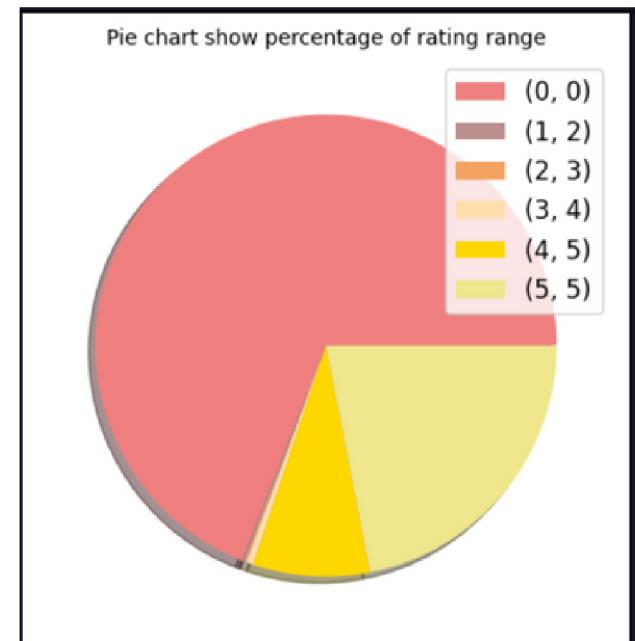
TRẢ LỜI CÂU HỎI

Câu 2: Lượt đánh giá trung bình, ảnh hưởng như thế nào?

Bước 2: Trực quan hóa dữ liệu để có được nhận xét
Nhận xét

- 0 sao chiếm đa số bộ dữ liệu.
- từ 4 đến 5 và 5 sao chiếm một
khoảng khá lớn trong số dữ liệu còn
lại
- còn 1 đến 4 sao dường như quá ít

Tại sao khách hàng lại không đánh giá, hoặc tại
sao sản phẩm lại không được đánh giá?



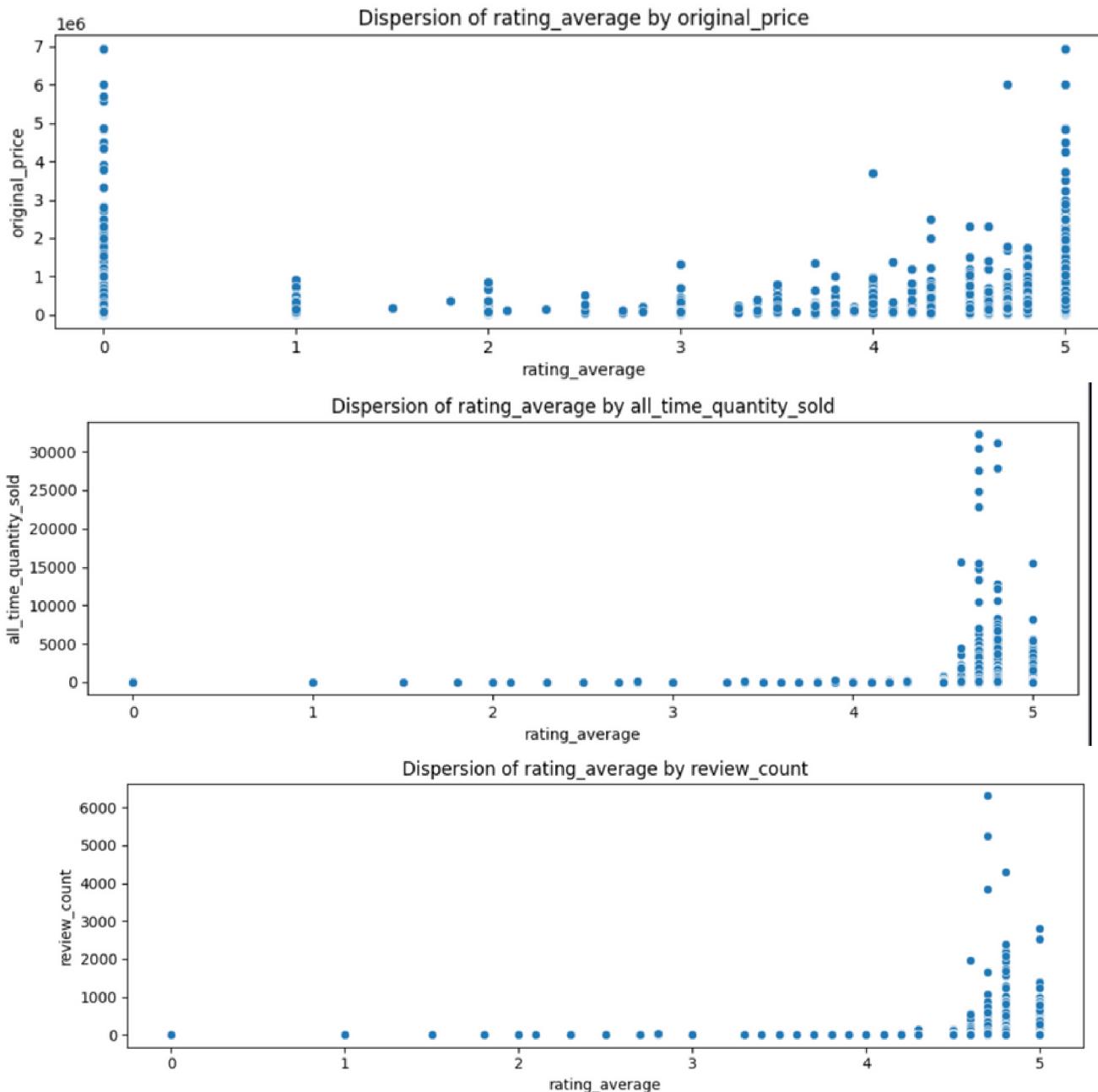
TRẢ LỜI CÂU HỎI

Câu 2: Lượt đánh giá trung bình, ảnh hưởng như thế nào?

Bước 2: Trực quan hóa dữ liệu để có được nhận xét

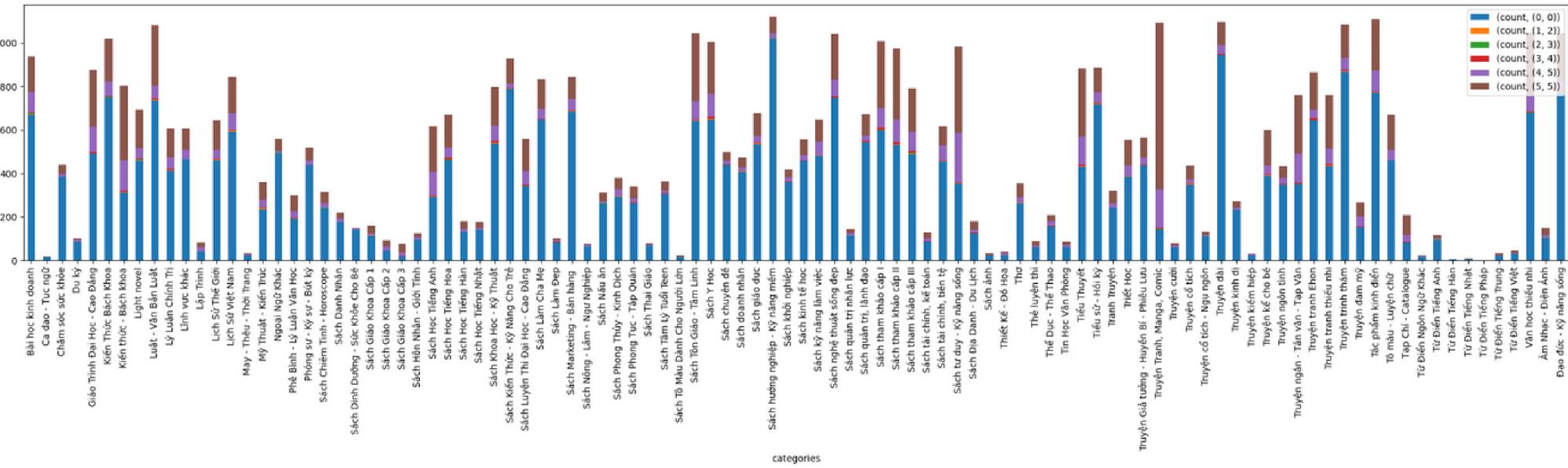
Xem sự phân tán của lượt đánh giá trung bình theo 3 trường dữ liệu: giá gốc, số lần bán, số lượt bình luận

TRẢ LỜI CÂU HỎI



TRẢ LỜI CÂU HỎI

Biểu đồ thể hiện lượt đánh giá của tất cả các loại sách.



TRẢ LỜI CÂU HỎI

Nhận xét

Với đồ thị theo giá gốc của các quyển sách, hầu như tất cả giá trị đều trong vùng từ 0 đến 300 ngàn đồng là chủ yếu. Tuy nhiên với độ đánh giá là '0 sao', hầu như số lượt mua hàng xấp xỉ hoặc hoàn toàn là 0 mặc dù sách ở mức đánh giá '0 sao' có tương đối nhiều sách được bán.

Việc có thể có được mức đánh giá trung bình cao nhưng lại khá ít đơn đặt hàng, theo em dự đoán rằng:

- + Việc mua hàng đối với các khách hàng đầu tiên đã xảy ra 1 vài trực trặc (như hư hỏng, không đúng như trong mô tả sản phẩm,...), dẫn tới sự kém tin tưởng của shop, nên dẫn đến mức đánh giá trung bình ngày một thấp đi. Số khách hàng đến sau dựa vào 1 số bình luận, số lượt đánh giá có trên tiki đưa ra quyết định mua hàng của mình.
- + Các quyển sách '0 sao' không được khách hàng tìm nhiều, hoặc không quá nổi tiếng, hoặc không cần thiết nên không có lượt đánh giá và bình luận.
- + Ngoài ra, thông tin được tìm kiếm ở câu 1, có một số cuốn sách bị trùng tên nhưng có nhiều shop bán. Việc một shop đầy sự tin tưởng trong suốt quá trình bán online trên tiki, dẫn đến việc khách hàng cũ sẽ ưu tiên mua sách ở shop quen thuộc hơn là mua ở một shop có lượt rating thấp.

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

Trả lời câu hỏi thành công:

- Nếu chúng ta biết được nhu cầu của khách hàng, việc sản xuất cho nhà nhà sản xuất sẽ thuận tiện hơn, đáp ứng được cung cầu của thị trường.
- Từ đó sẽ phát triển hơn trong việc phát hành sách

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

Khó khăn:

- Mỗi 1 thể loại thì sẽ có định nghĩa về edition và book_cover khác nhau. Ta cần chứng minh sự khác nhau đó.

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

```
view_book_cover = pd.DataFrame(df.book_cover.unique(), columns= ['Book Cover']).transpose()
```

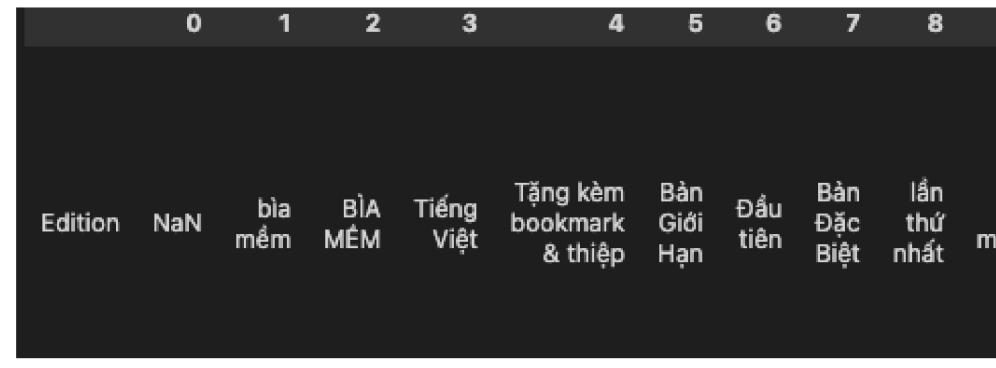
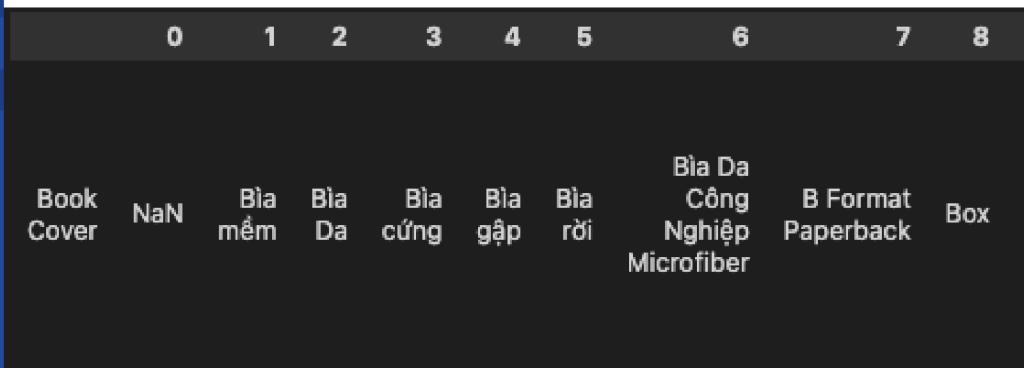
```
view_edition = pd.DataFrame(df.edition.unique(), columns= ['Edition']).transpose()
```



2 hàm bên trên dùng để thể hiện các loại bìa sách và phiên bản của 1 cuốn sách

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón



TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

Bước 1: Dữ liệu vẫn còn những giá trị Nan, ta sẽ chuyển thành None cho 2 loại cột

```
df.loc[df.edition.isnull(), 'edition'] = 'None'  
df.loc[df.book_cover.isnull(), 'book_cover'] = 'None'
```

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

Sau khi quan sát thì thấy trong edition lại chứa các phần tử bên book_cover như là bìa mềm, BÌA MỀM, ...

Ta sẽ thử quan sát 1 số loại category thử

```
LightNovel_view = get_df_by_categories('Light novel')
DuKy_view = get_df_by_categories('Du ký')
```

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

	Edition	Book Cover	Quantity Sold
0	Bản Giới Hạn	Bìa mềm	11
1	Bản Đặc Biệt	Bìa mềm	3
2	None	Bìa cứng	24
3	None	Bìa mềm	34948
4	None	Bìa rời	13
5	None	None	1961
6	Tặng kèm bookmark & thiệp	Bìa mềm	102
7	Đầu tiên	Bìa mềm	3

	Edition	Book Cover	Quantity Sold
0	BÌA MỀM	Bìa gấp	0
1	None	Bìa Da	19
2	None	Bìa cứng	13
3	None	Bìa mềm	927
4	None	None	787
5	Tiếng Việt	Bìa mềm	205
6	bìa mềm	Bìa gấp	0

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

Qua đây có thể thấy các `categories`, ta thấy mỗi thể loại được chia thành các `edition` với `book_cover` khác nhau.

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

Bước 2: Ta sẽ chọn thể loại Light novel để thực hiện quan sát dữ liệu

```
new_df = get_df_by_categories('Light novel')
```

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

Bước 3: Để quan sát được rõ ràng hơn về xu hướng, ta sẽ quan sát 2 nhóm sách:

- Nhóm bắt đầu bán từ năm 2021
- Nhóm bắt đầu bán từ năm 2022

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

Bước 3: Để quan sát được rõ ràng hơn về xu hướng, ta sẽ quan sát 2 nhóm sách:

```
new_df_2021 = new_df.loc[(new_df['day_ago_created'] > 365) & (new_df['day_ago_created'] <= 730)]
new_df_2022 = new_df.loc[new_df['day_ago_created'] <= 365]

plot_df_2022 = new_df_2022.groupby(['edition', 'book_cover'])['all_time_quantity_sold'].sum()
plot_df_2021 = new_df_2021.groupby(['edition', 'book_cover'])['all_time_quantity_sold'].sum()
```

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

Bước 4: Ta sẽ trực quan hóa bằng biểu đồ để tiện quan sát và đánh giá hơn:

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

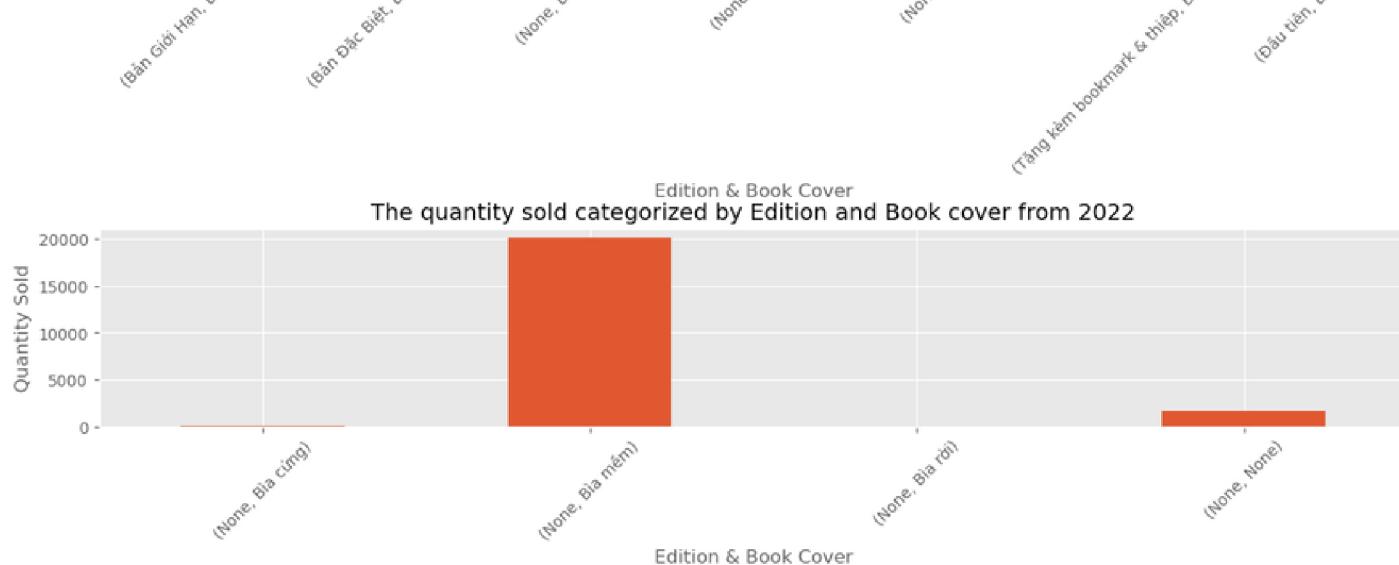
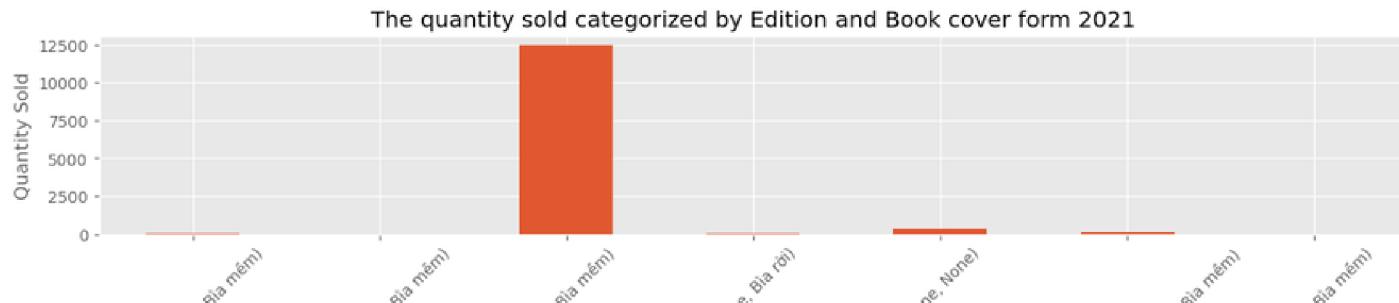
```
plt.style.use('ggplot')
fig, axes = plt.subplots(nrows = 2, ncols = 1, figsize=(15, 8))

plt.subplots_adjust(hspace = 1.5)
plot_df_2021.plot.bar(ax = axes[0], rot = 45)
axes[0].set_title('The quantity sold categorized by Edition and Book
cover form 2021')
axes[0].set(xlabel = 'Edition & Book Cover', ylabel = 'Quantity Sold')

plt.subplots_adjust(hspace = 1.5)
plot_df_2022.plot.bar(ax = axes[1], rot = 45)
axes[1].set_title('The quantity sold categorized by Edition and Book
cover from 2022')
axes[1].set(xlabel = 'Edition & Book Cover', ylabel = 'Quantity Sold')
```

TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón



TRẢ LỜI CÂU HỎI

Câu 3: Bìa sách và phiên bản của các loại sách có phải là mối quan tâm hàng đầu của khách hàng săn đón

Qua 2 biểu đồ thể hiện số lượng bán được từ năm 2021 và số lượng bán được từ năm 2022:

- Năm 2021, ta thấy có 5 loại `edition` và 4 loại `book_cover`. Có thể thấy khách hàng trên tiki có vẻ rất chuộng các loại sách loại thường và bìa mềm và các loại còn lại khách hàng mua số lượng không nhiều.
- Năm 2022, ta thấy chỉ còn lại 1 loại `edition` và vẫn còn 4 loại `book_cover`. Vậy qua 1 năm thì các loại sách nhập về không còn các `edition`, chỉ còn None (phiên bản thường). Các loại sách này nhập trong năm 2022 nhập về vẫn được các khách hàng vẫn chuộng các loại sách loại thường và bìa mềm và các loại còn lại vẫn không được nhiều

Vậy ta có thể suy đoán thử xem năm 2023, nhu cầu khách hàng vẫn là các sách bìa mềm với phiên bản thường chăng?

TRẢ LỜI CÂU HỎI

Câu 4: Các cuốn sách được nhiều sự quan tâm đánh giá từ khách hàng, có khuyến mãi như thế nào ?

Trả lời câu hỏi thành công:

Tìm hiểu được về việc đánh giá của khách hàng có ảnh hưởng đến giảm giá của shop không? từ đó shop có thể có ra thêm các loại event sale hợp lý để tăng sự tin tưởng của khách hàng và lấy được thêm các rating cao

TRẢ LỜI CÂU HỎI

Câu 4: Các cuốn sách được nhiều sự quan tâm đánh giá từ khách hàng, có khuyến mãi như thế nào ?

Khó khăn:

Quan sát toàn bộ số lượng sẽ dẫn đến chúng ta khó thấy sự khác biệt trong thay đổi tỉ lệ `discount_rate`. Ta sẽ số liệu thành thành sách được bán vào năm 2021 và năm 2022.

TRẢ LỜI CÂU HỎI

Câu 4: Các cuốn sách được nhiều sự quan tâm đánh giá từ khách hàng, có khuyến mãi như thế nào ?

Bước 1: Ta sẽ lấy dữ liệu là những cuốn sách bắt đầu được bán vào năm 2021 và vào năm 2022

```
df_2021 = df.loc[(df['day_ago_created'] > 365) & (new_df['day_ago_created'] <= 730)]  
df_2022 = df.loc[df['day_ago_created'] <= 365]
```

```
new_df_2021 = df_2021[['discount_rate', 'rating_average']]  
new_df_2022 = df_2022[['discount_rate', 'rating_average']]
```

TRẢ LỜI CÂU HỎI

Câu 4: Các cuốn sách được nhiều sự quan tâm đánh giá từ khách hàng, có khuyến mãi như thế nào ?

Bước 2: Ta sẽ đánh giá lại các mức ratings đã được chia ra như sau:

- (0,0) ứng với 0 sao
- (0,1) ứng với (0,1)
- (1,2) ứng với [1,2)
- (2,3) ứng với [2,3)
- (3,4) ứng với [3,4)
- (4,5) ứng với [4,5)
- (5,5) ứng với 5 sao

TRẢ LỜI CÂU HỎI

Câu 4: Các cuốn sách được nhiều sự quan tâm đánh giá từ khách hàng, có khuyến mãi như thế nào ?

Bước 2: Ta sẽ đánh giá lại các mức ratings đã được chia ra như sau:

```
temp_df_2021 = pd.DataFrame({"rating_range":  
new_df_2021.rating_average.apply(rating_range)})  
temp_df_2022 = pd.DataFrame({"rating_range":  
new_df_2022.rating_average.apply(rating_range)})  
  
new_df_2021 = pd.concat([new_df_2021, temp_df_2021], axis= 1)  
new_df_2022 = pd.concat([new_df_2022, temp_df_2022], axis= 1)
```

TRẢ LỜI CÂU HỎI

Câu 4: Các cuốn sách được nhiều sự quan tâm đánh giá từ khách hàng, có khuyến mãi như thế nào ?

Bước 3: Ta sẽ trực quan hóa dữ liệu để tiện quan sát hơn

```
new_df_2021 = new_df_2021.sort_values('rating_range')
new_df_2022 = new_df_2022.sort_values('rating_range')

fig, axes = plt.subplots(nrows = 2, ncols = 1, figsize=(15, 8))
plt.subplots_adjust(hspace = 1)

mean_2021 = new_df_2021.groupby(['rating_range'])['discount_rate'].mean()
mean_2021 = mean_2021.to_frame().rename(columns= {'discount_rate': 'Mean'}).reset_index()
mean_2022 = new_df_2022.groupby(['rating_range'])['discount_rate'].mean()
mean_2022 = mean_2022.to_frame().rename(columns= {'discount_rate': 'Mean'}).reset_index()

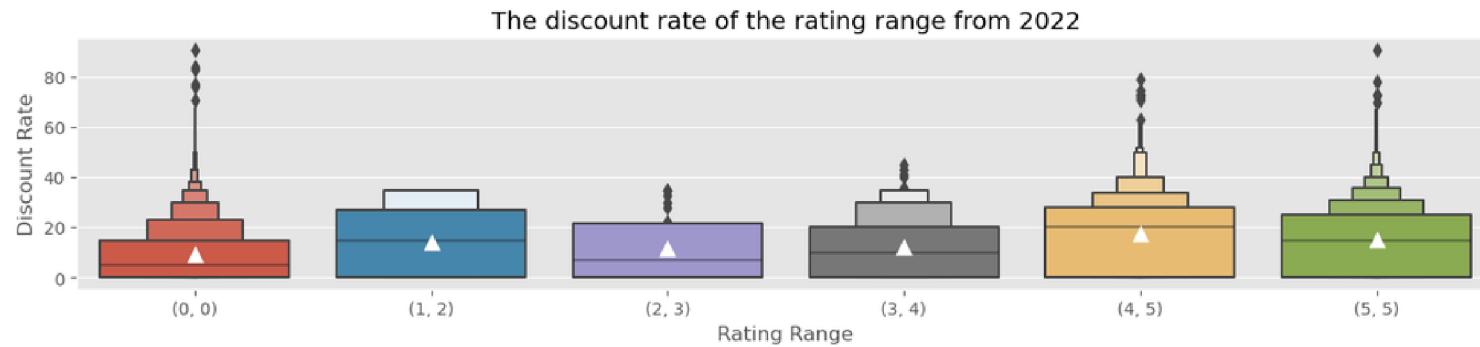
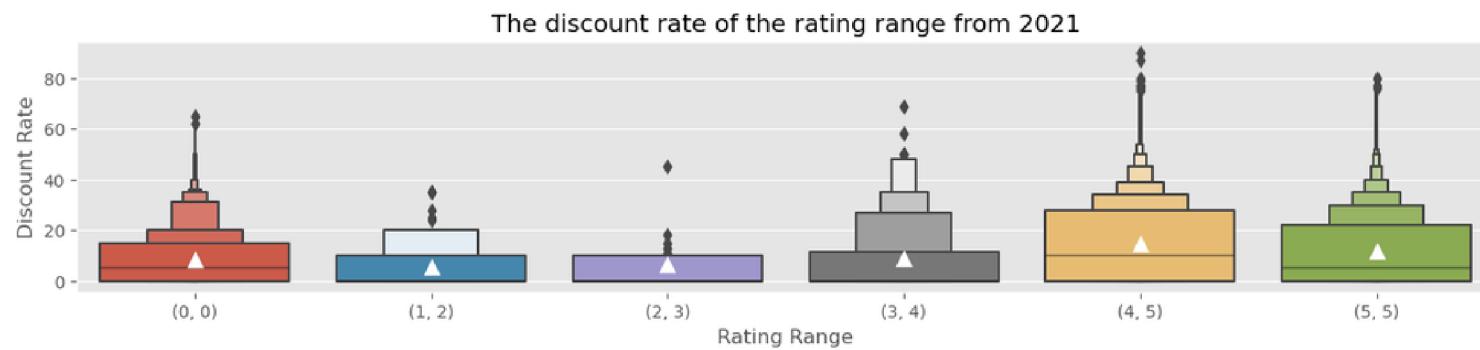
sns.swarmplot(data = mean_2021, x = 'rating_range', y = 'Mean', ax = axes[0], marker = '^', s = 10, palette= ['White'], dodge= True)
sns.swarmplot(data = mean_2022, x = 'rating_range', y = 'Mean', ax = axes[1], marker = '^', s = 10, palette= ['White'], dodge= True)

sns.boxenplot(data = new_df_2021, x = 'rating_range', y = 'discount_rate', ax = axes[0])
axes[0].set_title('The discount rate of the rating range from 2021')
axes[0].set(xlabel = 'Rating Range', ylabel = 'Discount Rate')

sns.boxenplot(data = new_df_2022, x = 'rating_range', y = 'discount_rate', ax = axes[1])
axes[1].set_title('The discount rate of the rating range from 2022')
axes[1].set(xlabel = 'Rating Range', ylabel = 'Discount Rate')
```

TRẢ LỜI CÂU HỎI

Câu 4: Các cuốn sách được nhiều sự quan tâm đánh giá từ khách hàng, có khuyến mãi như thế nào ?



TRẢ LỜI CÂU HỎI

Câu 4: Các cuốn sách được nhiều sự quan tâm đánh giá từ khách hàng, có khuyến mãi như thế nào ?

Qua 2 biểu đồ thể hiện sự giảm giá của các cuốn sách được bán từ 2 nhóm bán bắt đầu vào năm 2021 và vào năm 2022:

- Những cuốn sách được mở bán vào năm 2021 thì ta nhận thấy rằng về mức độ đa dạng về rating_range và discount_rate là có. Để ý những cuốn sách được đánh giá ở mức [4,5) sao thì thấy rằng đa dạng từ 0% cho đến 30% và trung bình rơi vào khoảng 15%, còn sách đánh giá 5* thì rơi tầm từ 0% cho đến 22% và mức trung bình rơi vào 12%. Và độ đa dạng đối với các sản phẩm không được đánh giá và đánh giá thấp trải giảm dần đều nhiều từ 0% đến 15%.
- Vào năm 2022 thì các cuốn sách có nhiều mức độ đa dạng về rating_range và discount_rate vẫn như năm 2021. Tuy nhiên tất cả mọi vùng đánh giá đều rơi vào trong tầm 0% đến 25% và tất cả đều có mức trung bình rơi vào 11% đến 16%. Vậy có vẻ như năm 2022 các shop đã nắm bắt được con số discount_rate mà dữ liệu đã đánh giá vào năm 2021 rồi chăng?

Vậy ta có thể dự đoán, các shop vẫn sẽ mở tiếp các ưu đãi cho các loại sách tầm 15% để tiếp tục thu hút các khách hàng vào 2023?

TRẢ LỜI CÂU HỎI

Câu 5: Số trang và bìa của 1 quyển sách có gây ảnh hưởng đến giá thành tiền của cuốn sách đó không?

Trả lời câu hỏi thành công:

Chúng ta sẽ phân biệt được giá thành của các loại sách, từ đó sẽ tiêu dùng hợp lý hơn vào sách.

TRẢ LỜI CÂU HỎI

Câu 5: Số trang và bìa của 1 quyển sách có gây ảnh hưởng đến giá thành tiền của cuốn sách đó không?

Khó khăn:

Những cột dữ liệu để phục vụ cho câu này vẫn còn cần được xử lý thêm

TRẢ LỜI CÂU HỎI

Câu 5: Số trang và bìa của 1 quyển sách có gây ảnh hưởng đến giá thành tiền của cuốn sách đó không?

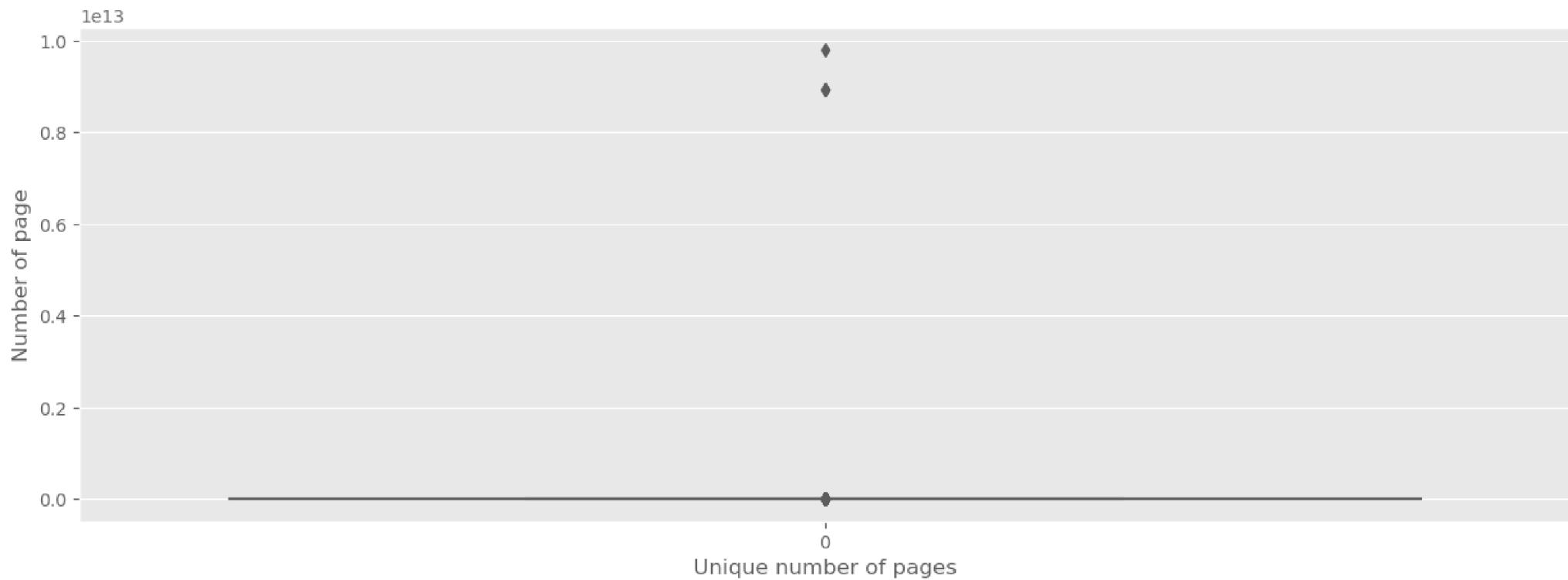
Bước 1: Ta sẽ kiểm tra xem là các trang sách có hợp lệ chưa? Ta sẽ dùng biểu đồ để thể hiện điều đó

```
page_df = df.number_of_page.unique()
plt.style.use('ggplot')
fig, ax = plt.subplots(figsize=(15, 5))

sns.boxplot(data = page_df)
ax.set(xlabel = 'Unique number of pages', ylabel = 'Number of page')
```

TRẢ LỜI CÂU HỎI

Câu 5: Số trang và bìa của 1 quyển sách có gây ảnh hưởng đến giá thành tiền của cuốn sách đó không?



TRẢ LỜI CÂU HỎI

Câu 5: Số trang và bìa của 1 quyển sách có gây ảnh hưởng đến giá thành tiền của cuốn sách đó không?

Qua đồ thị trên có thể rõ ràng thấy là điểm ngoại lai có giá trị quá lớn
 Theo thông tin, kiểm được trên google (link: <https://www.noron.vn/post/ban-co-biet-sach-va-nhung-ky-luc-thu-vi-40dxrlpfh9t4>) thì cuốn sách dày nhất chỉ có 5000 trang.

Dự đoán: có thể là do nhập lỗi thông số trang hay nhập thiếu?

TRẢ LỜI CÂU HỎI

Câu 5: Số trang và bìa của 1 quyển sách có gây ảnh hưởng đến giá thành tiền của cuốn sách đó không?

Bước 2: Ta sẽ loại bỏ những cuốn sách có trên 5000 trang và loại những cuốn sách 0 trang

```
new_df = df.loc[(df['number_of_page'] <= 5000) & (df['number_of_page'] != 0)]
```

TRẢ LỜI CÂU HỎI

Câu 5: Số trang và bìa của 1 quyển sách có gây ảnh hưởng đến giá thành tiền của cuốn sách đó không?

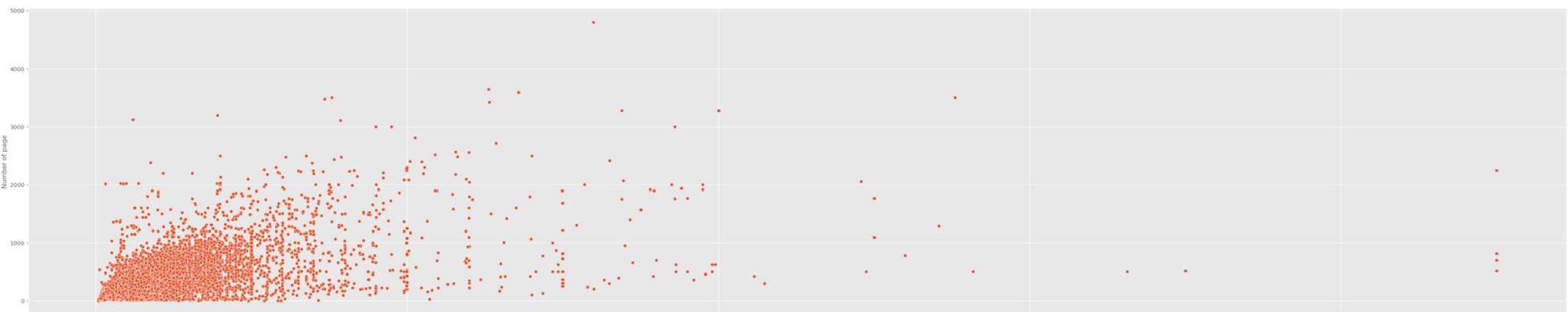
Bước 3: Ta sẽ minh họa mối liên hệ giữa giá tiền và số trang của sách

```
plt.style.use('ggplot')
fig, ax = plt.subplots(figsize=(50, 10))

sns.scatterplot(data = new_df[['number_of_page', 'original_price']], x = 'original_price', y = 'number_of_page')
ax.set(xlabel = 'Original price', ylabel = 'Number of page')
```

TRẢ LỜI CÂU HỎI

Câu 5: Số trang và bìa của 1 quyển sách có gây ảnh hưởng đến giá thành tiền của cuốn sách đó không?



TRẢ LỜI CÂU HỎI

Câu 5: Số trang và bìa của 1 quyển sách có gây ảnh hưởng đến giá thành tiền của cuốn sách đó không?

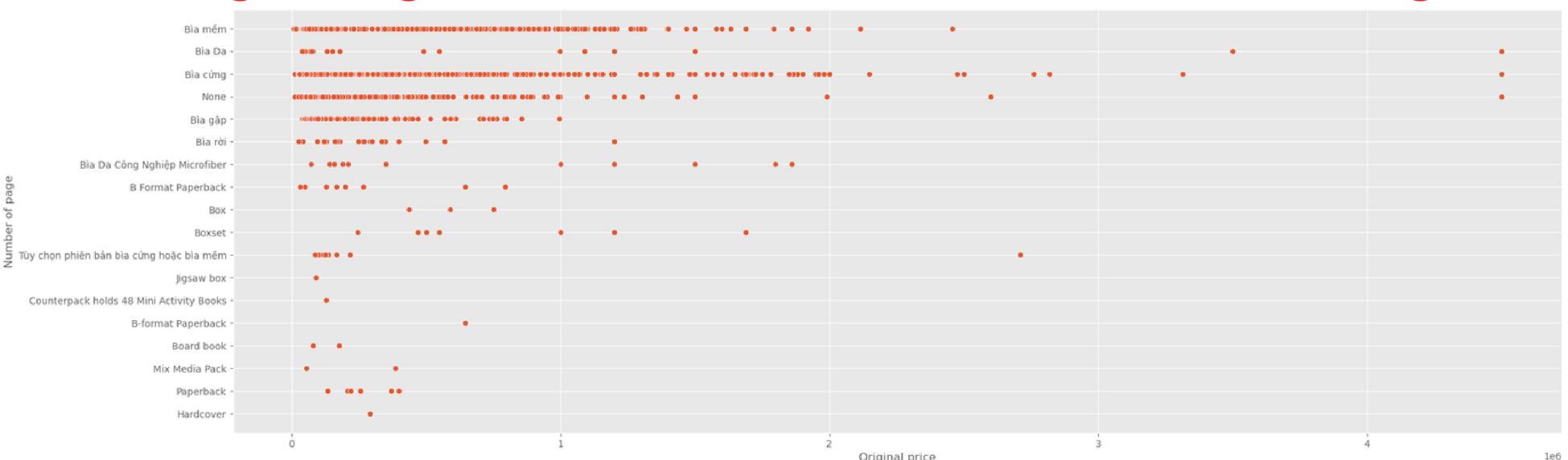
Bước 4: Ta sẽ minh họa mối liên hệ giữa giá tiền và bìa sách của sách

```
plt.style.use('ggplot')
fig, ax = plt.subplots(figsize=(50, 10))

sns.scatterplot(data = new_df[['number_of_page', 'original_price']], x = 'original_price', y =
'number_of_page')
ax.set(xlabel = 'Original price', ylabel = 'Number of page')
```

TRẢ LỜI CÂU HỎI

Câu 5: Số trang và bìa của 1 quyển sách có gây ảnh hưởng đến giá thành tiền của cuốn sách đó không?



TRẢ LỜI CÂU HỎI

Câu 5: Số trang và bìa của 1 quyển sách có gây ảnh hưởng đến giá thành tiền của cuốn sách đó không?

Qua 2 biểu đồ trên, ta có thể nhận xét rằng:

- Với biểu đồ thứ 1, ta thấy với những cuốn sách được bán ra tầm khoảng dưới 1000 trang thì giá tiền này độ khoảng 0 đồng cho đến 500 nghìn đồng. Ngoài ra, có vẻ như giá tiền hầu như không đồng biến với số trang, vẫn thấy chiều hướng nếu giá tiền càng mắc thì số trang tăng theo và vẫn có trường hợp số trang ít mà tiền vẫn cao (trường hợp nhiều trang thì giá tiền càng rẻ thì hầu như là không).
- Với biểu đồ thứ 2, ta thấy với những cuốn sách có 'bìa mềm, bìa cứng, không bìa, bìa gập' thì mức độ phân bố nhiều ở các giá tiền 500 nghìn đồng. Trên 500 nghìn đồng thì 'bìa gập và không bìa' có dấu hiệu thua dần trong khi 'bìa mềm và bìa cứng' vẫn phân bố dày đặc cho tới giá 2 triệu đồng. Hmm vậy có thể nói chất liệu 'bìa mềm và bìa cứng' có thể quyết định giá tiền của 1 cuốn sách chăng? Trong khi còn lại phân bố rời rạc không rõ được dấu hiệu để đánh giá. 😕

Vậy qua đó có thể thấy `original_price` có đi chung với `book_cover`, `number_of_page`. Từ đó ta có thể đánh giá cả của 1 cuốn sách có hợp lý hay không để mua hàng

TRẢ LỜI CÂU HỎI

Câu 6: xu hướng tình trạng của các sách vào ngày nay

Trả lời câu hỏi thành công:

Khách hàng sẽ biết được xu hướng tình trạng các món hàng như là còn hàng hay hết hàng hay chuẩn bị có những preorder nào. Từ đó, sẵn sàng tâm lý đi mua hàng, camp sách,... do tình trạng sách có thể hết hàng liên tục có khi không về hàng nữa? Từ đó cũng có thể dịch vụ camp sách lại phát triển?

TRẢ LỜI CÂU HỎI

Câu 6: xu hướng tình trạng của các sách vào ngày nay

Bước 1: Ta sẽ code hàm phân các sản phẩm theo từ ngày cách đây sản xuất thành theo năm nào có cuốn sách đó

```
def Year(x):
    if x <= 365:
        return '2022'
    elif x <= 730:
        return '2021'
    elif x <= 1095:
        return '2020'
    else:
        return '2019'
```

TRẢ LỜI CÂU HỎI

Câu 6: xu hướng tình trạng của các sách vào ngày nay

Bước 2: Chuẩn dataframe để cho bước trực quan hóa

```
new_df = df[['day_ago_created', 'inventory_type']]  
temp_df = pd.DataFrame({'year': new_df['day_ago_created'].apply(Year)})  
new_df = pd.concat([temp_df, new_df], axis = 1)  
new_df = new_df.loc[new_df['year'] != '2019']
```

TRẢ LỜI CÂU HỎI

Câu 6: xu hướng tình trạng của các sách vào ngày nay

Bước 2: Chuẩn bị dataframe để cho bước trực quan hóa

```
line_graph = new_df.groupby(['inventory_type'])  
['year'].value_counts().unstack(fill_value=0).stack().reset_index()  
line_graph.columns = [*line_graph.columns[:-1], 'quantity']
```

TRẢ LỜI CÂU HỎI

Câu 6: xu hướng tình trạng của các sách vào ngày nay

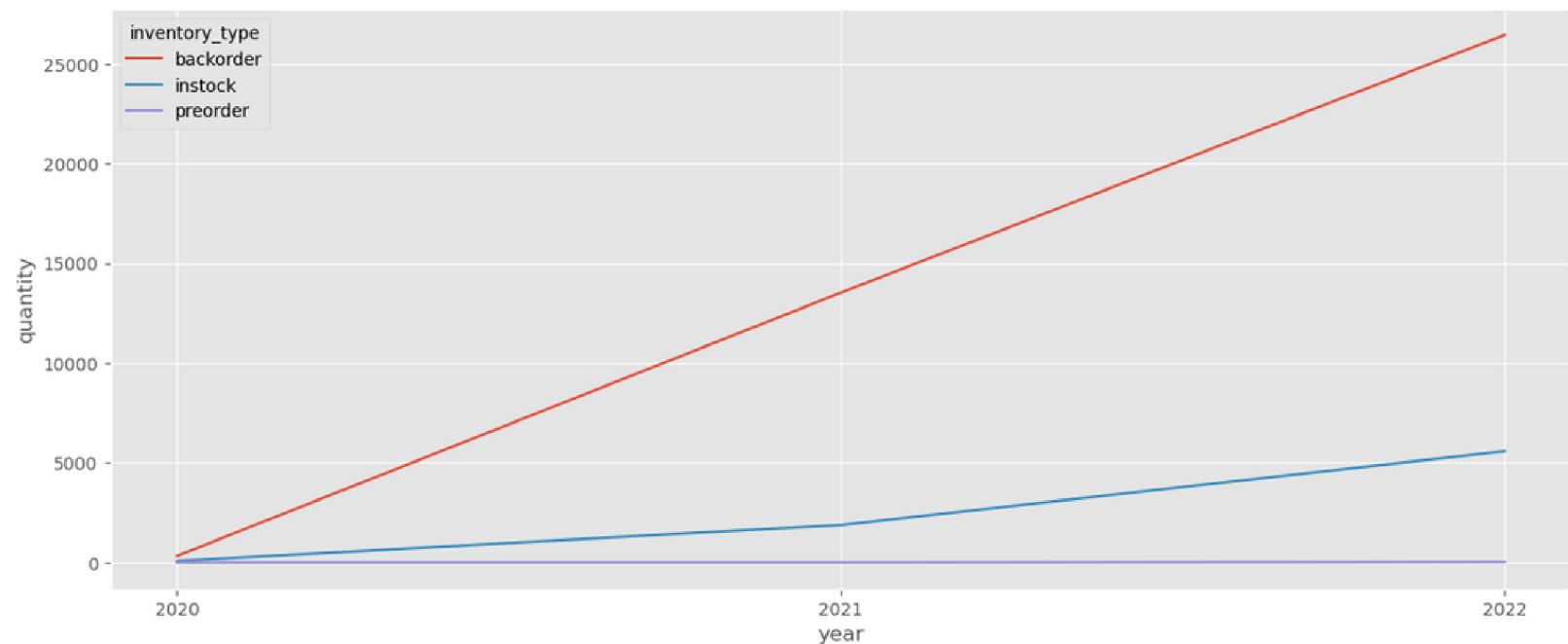
Bước 3: Ta sẽ trực quan hóa dữ liệu:

```
plt.style.use('ggplot')
fig, ax = plt.subplots(figsize=(15, 6))

sns.lineplot(ax = ax, data = line_graph, x = 'year', y = 'quantity', hue = 'inventory_type')
plt.show()
```

TRẢ LỜI CÂU HỎI

Câu 6: xu hướng tình trạng của các sách vào ngày nay



TRẢ LỜI CÂU HỎI

Câu 6: xu hướng tình trạng của các sách vào ngày nay

Bước 4: Tạo lại chuẩn bị thêm dataframe để cho bước trực quan hóa tiếp theo

```
pie_df_20 = line_graph.loc[line_graph['year'] == '2020']
pie_df_21 = line_graph.loc[line_graph['year'] == '2021']
pie_df_22 = line_graph.loc[line_graph['year'] == '2022']
```

TRẢ LỜI CÂU HỎI

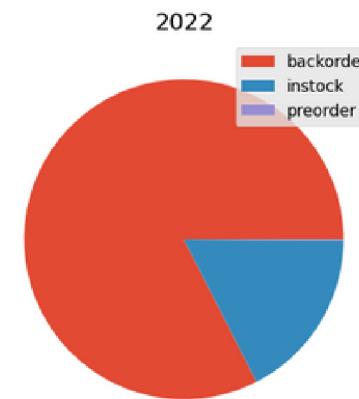
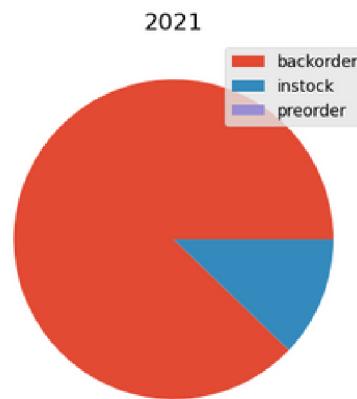
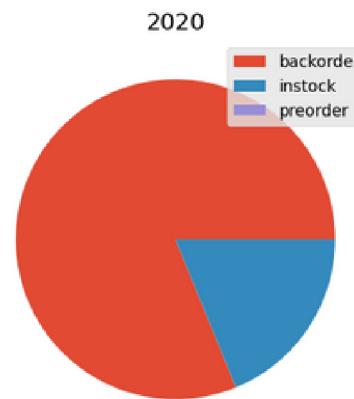
Câu 6: xu hướng tình trạng của các sách vào ngày nay

Bước 4: Ta sẽ trực quan hóa dữ liệu theo thêm kiểu khác để rõ hơn:

```
plt.style.use('ggplot')
fig, axes = plt.subplots(nrows = 1, ncols = 3, figsize = (15,15))
plt.subplots_adjust(hspace = 0.5)
axes[0].pie(pie_df_20['quantity'])
axes[0].legend(pie_df_20['inventory_type'])
axes[1].pie(pie_df_21['quantity'])
axes[1].legend(pie_df_21['inventory_type'])
axes[2].pie(pie_df_22['quantity'])
axes[2].legend(pie_df_22['inventory_type'])
plt.show()
```

TRẢ LỜI CÂU HỎI

Câu 6: xu hướng tình trạng của các sách vào ngày nay



TRẢ LỜI CÂU HỎI

Câu 6: xu hướng tình trạng của các sách vào ngày nay

Qua các biểu đồ line và pie ta thấy rằng:

- số lượng backorder (đã hết hàng và đang về lại hàng) thì ngày càng tăng vọt so qua các năm.
- Số lượng instock (còn hàng) thì cũng tăng nhẹ.
- Còn số lượng pre order thì chỉ mỗi từ năm 2022 có 27 sản phẩm và từ năm 2021, từ năm 2020 thì lại không có (chứng tỏ các sản phẩm được pre order đã được về không quá 1 năm :>)

Dự đoán được rằng các sản phẩm vào năm 2023 có thể hết hàng càng nhanh => khách hàng cần phải tranh nhau mua nhanh hơn, nhiều hiện tượng đứng camp sách hơn chăng? . Tuy nhiên thì số lượng tồn kho cũng tăng nhẹ thì vẫn đỡ phần nào.

MÔ HÌNH HOÁ DỮ LIỆU

- 1 Xác định câu hỏi cần trả lời
- 2 Phân tích, xử lý dữ liệu để chuẩn bị cho bước mô hình hóa
- 3 Huấn luyện mô hình
- 4 Đánh giá mô hình, tìm ra mô hình phù hợp nhất, tinh chỉnh các siêu tham số

MÔ HÌNH HÓA DỮ LIỆU

1. Xác định câu hỏi cần trả lời

Bài toán dự đoán discount_rate (phần trăm giảm giá) của một sản phẩm sách
dựa vào các thông tin của cuốn sách.

- Đây là một bài toán hồi quy
- Input là các đặc trưng ảnh hưởng tới tỉ lệ giảm giá của một cuốn sách
- Output là tỉ lệ giảm giá dự đoán của cuốn sách đó

MÔ HÌNH HÓA ĐỮ LIỆU

1. Xác định câu hỏi cần trả lời

Trả lời được câu hỏi này sẽ giúp cho cả khách hàng và người bán có được những lợi ích sau:

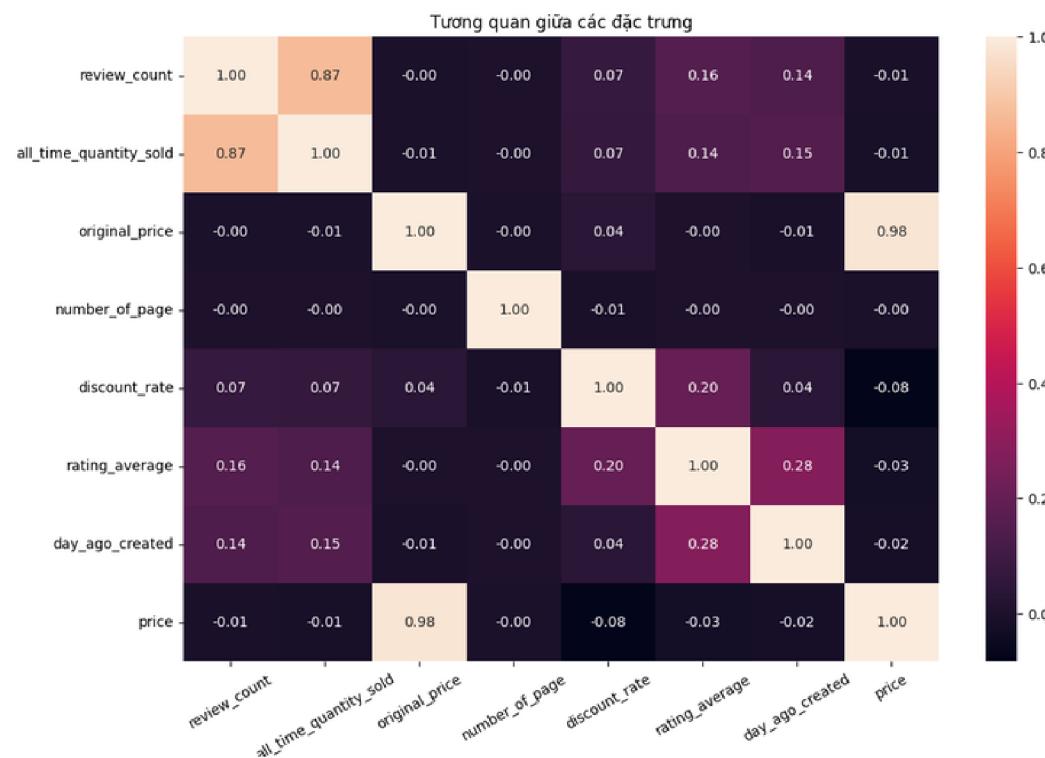
- Người bán quyết định được mức sale phù hợp cho từng sản phẩm, từ đó bán được nhiều sản phẩm hơn.
- Người mua chọn được sản phẩm giá rẻ, có tỉ lệ giảm giá cao, tối ưu được chi phí bỏ ra.

MÔ HÌNH HÓA DỮ LIỆU

2. Phân tích, xử lý dữ liệu để chuẩn bị cho bước mô hình hóa

- Kiểm tra kiểu dữ liệu của các cột, tạo một dataframe để xem min, max, phần trăm giá trị thiếu của những cột có kiểu dữ liệu số, một dataframe khác để xem tỉ lệ thiếu dữ liệu, các giá trị khác nhau của những cột có kiểu dữ liệu categorical (phân loại).
- Vẽ biểu đồ heatmap thể hiện tương quan giữa các đặc trưng, từ đó nhận xét các mối quan hệ của chúng.

MÔ HÌNH HÓA DỮ LIỆU



MÔ HÌNH HÓA DỮ LIỆU

2. Phân tích, xử lý dữ liệu để chuẩn bị cho bước mô hình hóa

Có thể thấy các mối quan hệ như sau:

- all_time_quantity_sold và review_count: khá hợp lý, tổng lượng bán cao thì số lượng review cũng cao
- original_price và price: giá gốc cao thì giá sau khi giảm giá cũng cao

MÔ HÌNH HÓA DỮ LIỆU

2. Phân tích, xử lý dữ liệu để chuẩn bị cho bước mô hình hóa

Tiếp theo, ta chọn các đặc trưng phù hợp/ có ảnh hưởng đến kết quả làm input cho bài toán

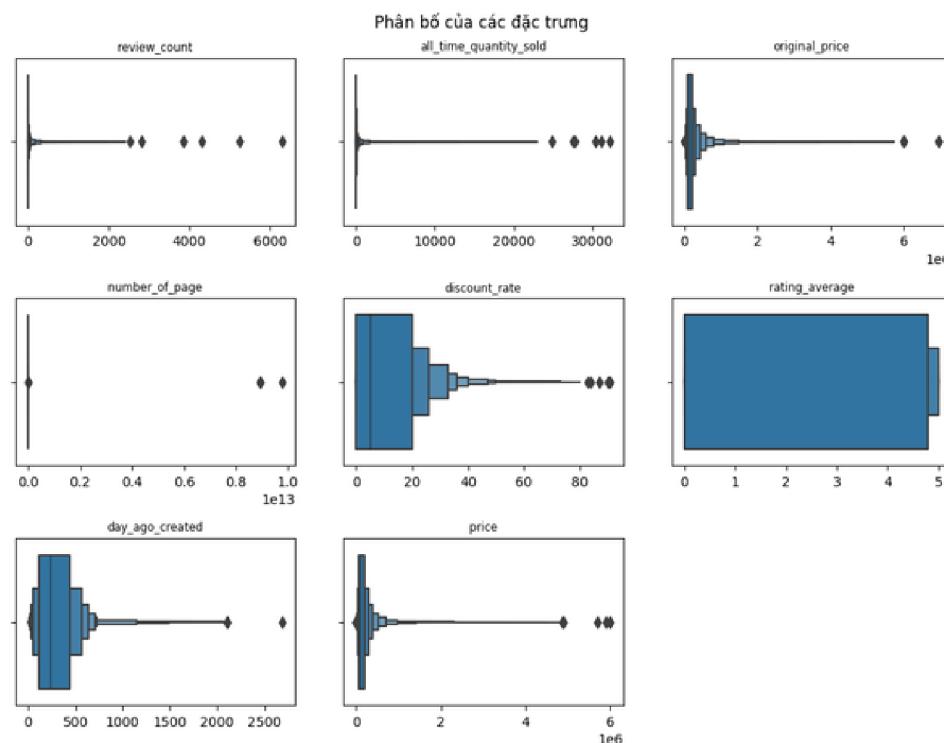
- Các cột số: original_price, rating_average, review_count, day_ago_created, all_time_quantity_sold, number_of_page
- Các cột phân loại: book_cover, categories, inventory_type, manufacturer

MÔ HÌNH HÓA DỮ LIỆU

2. Phân tích, xử lý dữ liệu để chuẩn bị cho bước mô hình hóa

Vẽ boxplot để xem phân bố của dữ liệu, từ đó điều chỉnh cho phù hợp

MÔ HÌNH HÓA DỮ LIỆU



MÔ HÌNH HÓA DỮ LIỆU

2. Phân tích, xử lý dữ liệu để chuẩn bị cho bước mô hình hóa

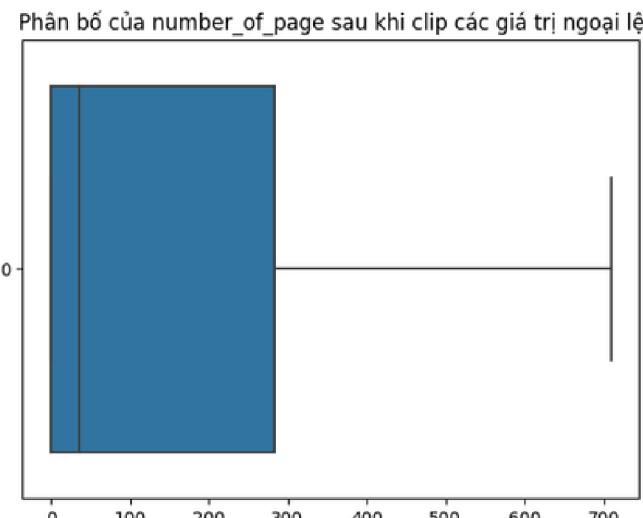
Nhận xét về phân bố của dữ liệu:

- review_count có phân bố chủ yếu ở giá trị 0, xấp xỉ 0 và rải rác đến 6000
- all_time_quantity_sold cũng tương tự, tuy nhiên các giá trị thua đến 30000
- original_price thì phân bố tập trung trong khoảng 0 đến 2e6
- discount_rate chủ yếu phân bố trong từ 0-40 và rải rác ở mức lớn hơn
- rating_average phân bố khá đồng đều trong miền giá trị
- day_ago_created tập trung trong khoảng 0 đến dưới 1000
- price phân bố gần như giống hệt original_price
- Riêng cột number_of_page có outliers khá "khủng" nên ta sẽ xử lý riêng

MÔ HÌNH HÓA DỮ LIỆU

2. Phân tích, xử lý dữ liệu để chuẩn bị cho bước mô hình hóa

Xử lý outliers của number_of_page: clip các giá trị outliers về các giá trị min, max của box plot bằng hàm BoxplotOutlierClipper()



MÔ HÌNH HÓA DỮ LIỆU

2. Phân tích, xử lý dữ liệu để chuẩn bị cho bước mô hình hóa

Tạo pipeline cho bước tiền xử lý các giá trị:

- Các cột categorical thì điền các giá trị thiếu bằng giá trị có tần xuất xuất hiện cao nhất rồi encode dưới dạng one-hot vector
- Các cột kiểu dữ liệu là số thì ta điền các giá trị thiếu bằng thuật toán KNN (K-nearest Neighbor): thuật toán này dựa vào các điểm gần nhất để nội suy giá trị cần tìm. Tiếp đó, scale các giá trị bằng bộ scale MinMaxScaler của scikit-learn

MÔ HÌNH HÓA DỮ LIỆU

3. Huấn luyện mô hình

Xác định độ đo đánh giá chung là MSE (mean square error) và MAE (mean absolute error)

Mô hình LinerRegression:

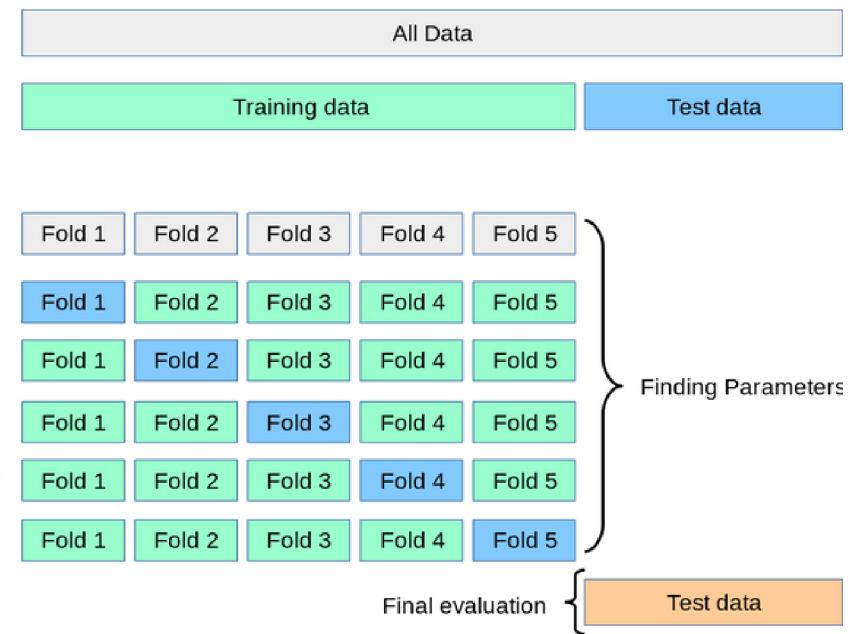
- Là mô hình đơn giản nhất của một bài toán hồi quy.
- Đối với dữ liệu ta đang sử dụng, mô hình này cho kết quả đánh giá khá tốt với độ lỗi $MSE = 108.656$, $MAE = 8.093$ trên tập train và $MSE = 109.089$, $MAE = 8.093$ trên tập test.

MÔ HÌNH HÓA DỮ LIỆU

3. Huấn luyện mô hình

Đánh giá chéo (Cross validation):

- Thủ tục lấy mẫu đặc biệt, dùng để đánh giá các mô hình machine learning trong quá trình huấn luyện
- Ta sẽ chia tập train ra k-folds không chồng lấn, có kích thước bằng nhau.
- Tại mỗi lượt huấn luyện, ta sẽ huấn luyện trên $(k-1)$ folds và đánh giá trên fold còn lại
- Đánh giá được khả năng dự báo đối với dữ liệu mà mô hình chưa nhìn thấy



MÔ HÌNH HÓA DỮ LIỆU

3. Huấn luyện mô hình

Áp dụng Cross validation lên mô hình LinerRegression:

- Sử dụng class KFold() với n_splits là số lần chia dữ liệu, shuffle là xáo trộn dữ liệu hay không.
- Sử dụng độ đo "neg_mean_absolute_error"
- Kết quả trên tập train là: Mean MAE = -8.152 và Std MAE = 0.057

MÔ HÌNH HÓA DỮ LIỆU

4. Đánh giá mô hình, tìm ra mô hình phù hợp nhất, tinh chỉnh các siêu tham số

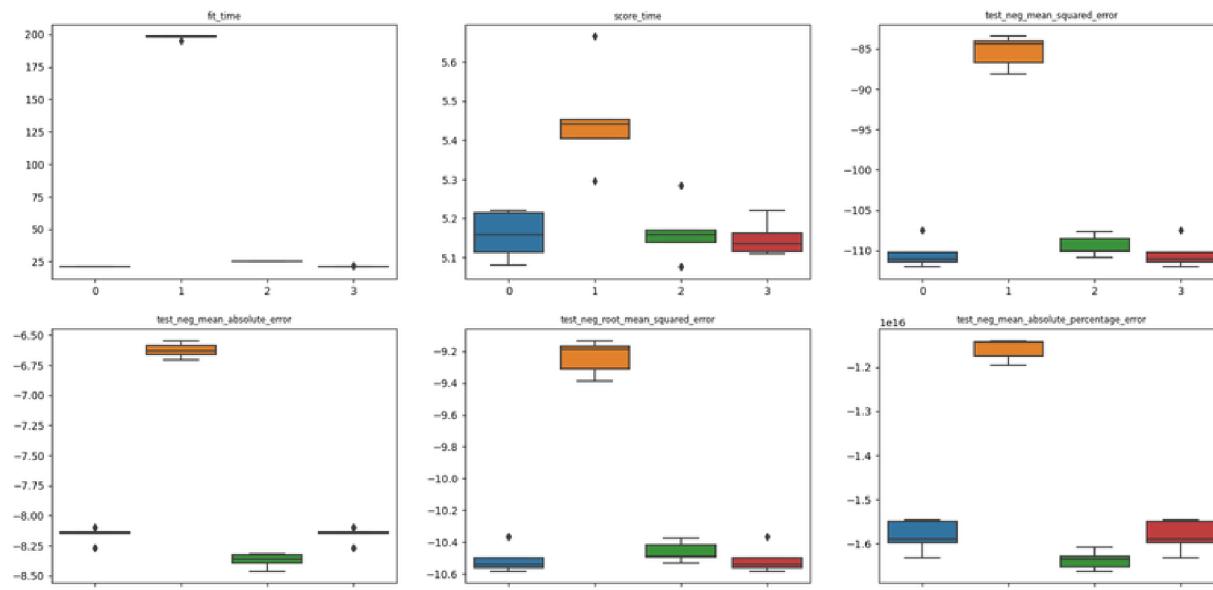
Đánh giá nhiều mô hình với các metrics khác nhau:

- Ta sử dụng 4 mô hình sau: `LinearRegression()`, `RandomForestRegressor()`, `GradientBoostingRegressor()`, `TransformedTargetRegressor()`
- Metrics dùng để đánh giá: MAE, MSE, RMSE, MAPE
- Lần lượt lặp qua 4 mô hình kết hợp với sử dụng cross validation để đánh giá độ lỗi của từng mô hình, từ đó rút ra kết luận nên sử dụng mô hình nào.

MÔ HÌNH HÓA DỮ LIỆU

4. Đánh giá mô hình, tìm ra mô hình phù hợp nhất, tinh chỉnh các siêu tham số

Scores Metrics



MÔ HÌNH HÓA DỮ LIỆU

4. Đánh giá mô hình, tìm ra mô hình phù hợp nhất, tinh chỉnh các siêu tham số

Dựa vào đồ thị boxplot minh họa, ta có thể thấy Random Forest Regression cho độ lỗi tốt nhất trong các mô hình, tuy nhiên fit_time và score_time cũng cao hơn

Vậy ta sẽ chọn mô hình Random Forest cho bài toán

- Thử nghiệm lại với mô hình này, ta được độ lỗi trên tập train là MSE = 12.343, MAE = 2.447; tập test là MSE= 81.452, MAE = 6.466
- Độ lỗi đã cải thiện rất nhiều so với các mô hình trên

MÔ HÌNH HÓA DỮ LIỆU

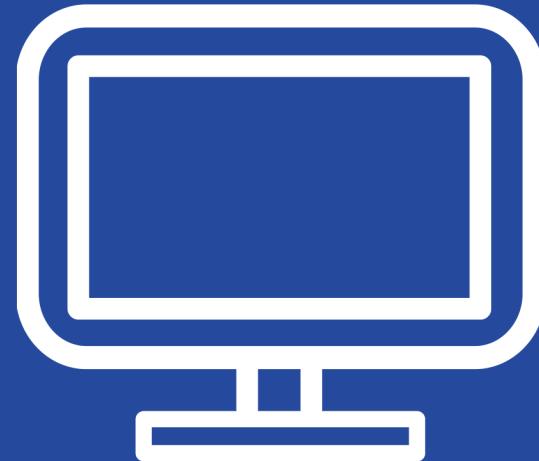
4. Đánh giá mô hình, tìm ra mô hình phù hợp nhất, tinh chỉnh các siêu tham số

Fine-tuning process: Kỹ thuật tinh chỉnh các siêu tham số

- Sử dụng kỹ thuật Randomized search CV để tìm ra bộ tham số phù hợp nhất cho mô hình.
- Class RadomizedSearchCV() của scikit-learn: Tìm kiếm ngẫu nhiên các bộ tham số trên không gian tham số, kết hợp với Cross-validation để tìm ra bộ tham số tốt nhất.

```
Best Parameters: {'n_estimators': 1000, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'auto', 'max_depth': 120, 'bootstrap': True}
```

Kết quả này cho độ lỗi là tốt nhất (trong không gian tham số tự định nghĩa), tuy nhiên không tối ưu về mặt thời gian và độ lỗi cũng không cải thiện nhiều. Vì vậy, em sẽ chọn thuật toán Random Forest với các tham số mặc định (độ lỗi chấp nhận được và xấp xỉ với trường hợp tối ưu)



T H A N K Y O U !