

**ĐẠI HỌC HUẾ**

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

**🙢**

**BÁO CÁO ĐỒ ÁN**

**Học kỳ I, năm học 2021 - 2022**

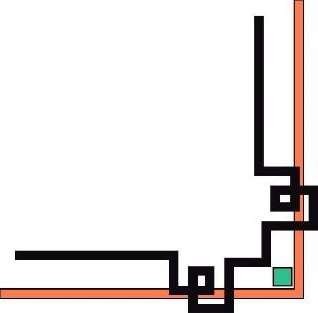
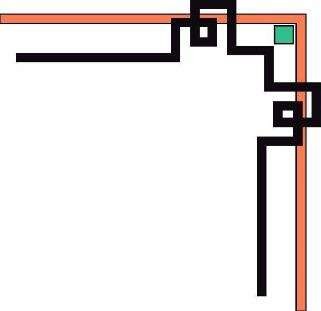
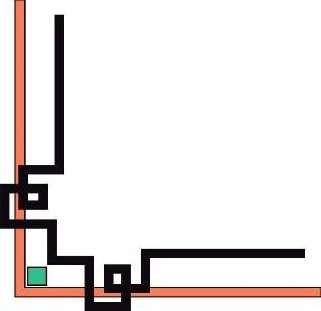
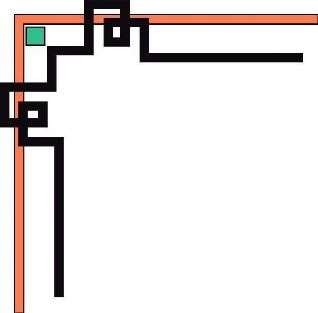
**Học phần:**

**PHÂN TÍCH DỮ LIỆU - R**

**Số phách**

*(Do hội đồng chấm thi ghi)*

**Thừa Thiên Huế, tháng 01 năm 2024**



**ĐẠI HỌC HUẾ**

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

**🙢**

**BÁO CÁO ĐỒ ÁN**

**Học kỳ I, năm học 2021 - 2022**

**Học phần:**

**PHÂN TÍCH DỮ LIỆU - R**

**Giảng viên hướng dẫn:** TS. Hồ Quốc Dũng

**Lớp:** Khoa học dữ liệu và trí tuệ nhân tạo - K3

**Sinh viên thực hiện:** Nguyễn Văn Minh Khánh

*(ký và ghi rõ họ tên)*

**Số phách**

*(Do hội đồng chấm thi ghi)*

**Thừa Thiên Huế, tháng 01 năm 2024**

# LỜI CẢM ƠN

“Em đã rất cố gắng và nỗ lực trong bài báo cáo đồ án này. Tuy nhiên, sẽ không thể thực hiện được nếu không có sự hỗ trợ, giúp đỡ ân cần của giảng viên bộ môn – Phân tích dữ liệu R cũng như Ban giám hiệu Khoa Kỹ thuật và Công nghệ - Đại học Huế vì đã tạo điều kiện về cơ sở vật chất, như môi trường học tập thân thiện, giúp em phát huy hết khả năng học tập và rèn luyện nhân cách một cách hiệu quả.

Em muốn bày tỏ lòng biết ơn chân thành đối với giảng viên bộ môn và toàn thể giáo viên của Khoa Kỹ thuật và Công nghệ - Đại học Huế.

Em muốn bày tỏ lòng biết ơn đến gia đình và bạn bè vì đã luôn đồng hành, động viên và quan tâm em trên con đường học tập và trong cuộc sống.

Và lời cảm ơn đặc biệt cuối cùng em xin dành tặng cho bản thân chính mình vì đã không bỏ cuộc vào những lúc bản thân suy sụp, mệt mỏi nhất, cảm ơn bản thân đã luôn cố gắng để vượt qua những khó khăn tưởng chừng không thể bước tiếp, cảm ơn vì tất cả.”

# DANH MỤC HÌNH ẢNH

[Hình 1: Bảng DevTools 6](#_Toc155751877)

[Hình 2: Network > Fetch/XHR 6](#_Toc155751878)

[Hình 3: litst request - crawl id product 7](#_Toc155751879)

[Hình 4: Headers và params - crawl id product 7](#_Toc155751880)

[Hình 5: API - crawl id product 8](#_Toc155751881)

[Hình 6: Thông tin sản phẩm cần crawl 9](#_Toc155751882)

[Hình 7: Tổng quan về tập dữ liệu 13](#_Toc155751883)

[Hình 8: Kết quả hiển thị thư mục làm việc 14](#_Toc155751884)

[Hình 9: Số hàng và số cột của tập dữ liệu 14](#_Toc155751885)

[Hình 10: Trả về tên các trường dữ liệu 15](#_Toc155751886)

[Hình 11: Dataframe mới 15](#_Toc155751887)

[Hình 12: Các dòng bị trùng lặp 16](#_Toc155751888)

[Hình 13: Hiển thị lại DataFrame không chứa dòng trùng lặp 16](#_Toc155751889)

[Hình 14: Thống kê mô tả tập dữ liệu 17](#_Toc155751890)

[Hình 15: Biểu đồ cột 18](#_Toc155751891)

[Hình 16: Biểu đồ tròn 20](#_Toc155751892)

[Hình 17: Biểu đồ histogram 20](#_Toc155751893)

[Hình 18: Kết quả kiểm định t 21](#_Toc155751894)

[Hình 19: Kết quả kiểm tra giả thuyết 23](#_Toc155751895)

[Hình 20: Tóm tắt mô hình Linear Regression 23](#_Toc155751896)

[Hình 21: Biểu đồ Linear Regression 25](#_Toc155751897)

[Hình 22: Thông tin phân cụm dữ liệu 26](#_Toc155751898)

[Hình 23: Biểu đồ KMeans 27](#_Toc155751899)

[Hình 24: Kết quả PCA 28](#_Toc155751900)

[Hình 25: Biểu đồ trực quan PCA 29](#_Toc155751901)

# DANH MỤC BẢNG BIỂU

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc155752198)

[DANH MỤC HÌNH ẢNH ii](#_Toc155752199)

[DANH MỤC BẢNG BIỂU iii](#_Toc155752200)

[MỤC LỤC iv](#_Toc155752201)

[CHƯƠNG 1: MÔ TẢ KỸ THUẬT CRAWL DỮ LIỆU 5](#_Toc155752202)

[1.1 Kỹ thuật crawl dữ liệu 5](#_Toc155752203)

[1.1.1 Crawl id sản phẩm từ danh mục thời trang nam 5](#_Toc155752204)

[Crawl thông tin sản phẩm từ id 9](#_Toc155752205)

[1.2 Giới thiệu về tập dữ liệu đã crawl 13](#_Toc155752206)

[CHƯƠNG 2: PHÂN TÍCH TỔNG QUAN VỀ DỮ LIỆU 14](#_Toc155752207)

[2.1 Đọc dữ liệu 14](#_Toc155752208)

[2.1.1 Nhập các thư viện cần thiết 14](#_Toc155752209)

[2.1.2 Thông tin tập dữ liệu 14](#_Toc155752210)

[2.2 Tiền xử lý dữ liệu 15](#_Toc155752211)

[2.3 Thống kê mô tả 17](#_Toc155752212)

[2.4 Trực quan hóa dữ liệu 18](#_Toc155752213)

[2.4.1 Biểu đồ cột 18](#_Toc155752214)

[2.4.2 Biểu đồ tròn 19](#_Toc155752215)

[2.4.2 Biểu đồ histogram 20](#_Toc155752216)

[CHƯƠNG 3: CHỨNG MINH VÀ PHÂN TÍCH CÁC GIẢ THUYẾT, BÀI TOÁN 21](#_Toc155752217)

[3.1 Chứng minh giả thuyết (Hypothesis Testing) 21](#_Toc155752218)

[3.2 Dự đoán 23](#_Toc155752219)

[3.2.1 Linear Regression 23](#_Toc155752220)

[3.2.2 K-Means 25](#_Toc155752221)

[3.3 Phân tích PCA 28](#_Toc155752222)

[LINK MÃ NGUỒN 30](#_Toc155752223)

[TÀI LIỆU THAM KHẢO 31](#_Toc155752224)

[KẾT QUẢ KIỂM TRA ĐẠO VĂN 32](#_Toc155752225)

# CHƯƠNG 1: MÔ TẢ KỸ THUẬT CRAWL DỮ LIỆU

## 1.1 Kỹ thuật crawl dữ liệu

API (Application Programming Interface), là 1 giao diện mà giúp chúng ta thu thập và gửi dữ liệu sử dụng code. Chúng ta sử dụng APIs nhiều nhất khi thu thập dữ liệu, và điều đó sẽ được đề cập đến trong bài đồ án này.

Khi chúng ta muốn crawl dữ liệu từ API, chúng ta cần tạo 1 request. Requests được sử dụng ở hầu hết mọi website. Ví dụ, khi bạn truy cập một trang web nào đó, trình duyệt của bạn gửi 1 request đến server của trang web cần crawl. API request hoạt động với cách thức chính xác như vậy: bạn tạo 1 request đến API server để lấy dữ liệu, và server trả về cho bạn thứ bạn muốn.

Ở [chương 1](#_CHƯƠNG_1:_) này, chúng ta sẽ crawl dữ liệu từ trang web [tiki.vn](https://tiki.vn/), chủ yếu để lấy danh sách các sản phẩm (id, tên sản phẩm, giá cả,...) và lưu chúng vào một tệp CSV. Để thực hiện một yêu cầu (request) API, chúng ta sẽ sử dụng ngôn ngữ lập trình là Python để hỗ trợ gửi HTTP requests.

### 1.1.1 Crawl id sản phẩm từ danh mục thời trang nam

Để lấy được những thông tin đó thì đầu tiên chúng ta phải cần crawl id của từng sản phẩm trước.

Link trang web crawl id sản phẩm: <https://tiki.vn/thoi-trang-nam/c915>.

Nhập các thư viện cần thiết.

|  |
| --- |
| import requests # gửi HTTP requests và nhận HTTP responses  import time # tạo độ trễ giữa các requests  import random   # tạo ngẫu nhiên thời gian chờ  import pandas as pd # làm việc với dữ liệu dạng bản |

Tiếp đến chúng ta cần nhập header và params của trang web, để lấy được các thông số trong header và params. Ta cần làm theo các bước như sau:

Nhấn chuột phải 🡪 Kiểm tra hoặc nhấn phím F12 để mở bảng DevTools.

A screenshot of a computer

Description automatically generated

Hình : Bảng DevTools

Từ bảng DevTools, chọn Network 🡪 tick vào ô Fetch/XHR để hiện các list request (nếu không hiện thì tải lại trang)

A screenshot of a computer

Description automatically generated

Hình : Network > Fetch/XHR

Chọn list request chứa id sản phẩm, nhấn mũi tên bên trái của key data 🡪 xuất hiện các list id sản phẩm.

A screenshot of a computer

Description automatically generated

Hình : litst request - crawl id product

Để crawl được id chúng ta phải lấy thông tin header và params cũng như API của trang web.

A screenshot of a computer

Description automatically generated

Hình : Headers và params - crawl id product

Các thông số nằm ở phía dưới. Còn để lấy API thì nhấn đúp vào list request chứa id 🡪 xuất hiện cửa số mới 🡪 copy lại API nằm trên thanh địa chỉ.



Hình : API - crawl id product

Bắt đầu crawl dữ liệu, ở đây ta sẽ lấy dữ liệu từ 10 trang.

|  |
| --- |
| eaders = {      'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36',      'Accept': 'application/json, text/plain, \*/\*',      'Accept-Language': 'vi-VN,vi;q=0.9,fr-FR;q=0.8,fr;q=0.7,en-US;q=0.6,en;q=0.5',      'Referer': 'https://tiki.vn/thoi-trang-nam/c915',      'x-guest-token': '8YcN0C1W8zP6ElLJkTgdAMyxrpvQwb94q',      'Connection': 'keep-alive',      'TE': 'Trailers',  }  params = {      'limit': '40',      'include': 'sale-attrs,badges,product\_links,brand,category,stock\_item,advertisement',      'aggregations': '2',      'trackity\_id': '7e2ddf92-8de0-c763-991f-c6e7161b2dac',      'category': '915',      'page': '1',      'src': 'c915',      'urlKey':  'thoi-trang-nam',  }  product\_id = []  for i in range(1, 11):      params['page'] = i      response = requests.get('https://tiki.vn/api/personalish/v1/blocks/listings', headers=headers, params=params)#, cookies=cookies)      if response.status\_code == 200:          print('Request success!!!')          for record in response.json().get('data'):              product\_id.append({'id': record.get('id')})      time.sleep(random.randrange(3, 10)) |

Sau khi crawl thành công thì lưu id sản phẩm vào file csv.

|  |
| --- |
| df = pd.DataFrame(product\_id)  df.to\_csv('crawl\_product\_id.csv', index=False) |

### Crawl thông tin sản phẩm từ id

Để crawl được thông tin sản phẩm thì ta nhấn chọn một sản phẩm bất kì để chuyển sang trang mới.

Mở DevTools rồi làm như các bước cũ. Lúc này phải chọn request chứa thông tin sản phẩm như bên dưới. Ở phần Preview sẽ chứa các trường dữ liệu cần crawl (giá cả, tên sản phẩm, đánh giá,…)

A screenshot of a computer

Description automatically generated

Hình : Thông tin sản phẩm cần crawl

Phương thước lấy API, headers, params và cookies như bước crawl id.

Điều chỉnh thông số cho cookie, headers và params của trang.

|  |
| --- |
| import pandas as pd  import requests  import time  import random  from tqdm import tqdm  cookies = {      'TIKI\_RECOMMENDATION': '6daf0ec0d8aeabb3461783efada3ac72',      'TKSESSID': 'b971a21329324f186bf7f28716e4b6cf',      'TOKENS': '{%22access\_token%22:%22YcN0C1W8zP6ElLJkTgdAMyxrpvQwb94q%22}',      '\_\_IP': '1746718024',      '\_\_R': '0',      '\_\_iid': '749',      '\_\_iid': '749',      '\_\_su': '0',      '\_\_su': '0',      '\_\_tb': '0',      '\_\_uidac': '01659576d001836f069b27e32d7a7499',      '\_\_uif': '\_\_uid%3A7642940961746729271%7C\_\_ui%3A-1%7C\_\_create%3A1704294096',      '\_\_utm': 'source%3Dtiki-aff%7Cmedium%3Dtiki-aff%7Ccampaign%3DAFF\_NBR\_TIKIAFF\_UNK\_TIKIVN-D3K72LS2\_ALL\_VN\_ALL\_UNK\_UNK\_TAPX.f1bf430a-65cf-418a-b43e-28dfa9168045\_TAPU.e8811fd7-587b-4b1d-b685-d9586955086f',      '\_\_utm': 'source%3Dtiki-aff%7Cmedium%3Dtiki-aff%7Ccampaign%3DAFF\_NBR\_TIKIAFF\_UNK\_TIKIVN-D3K72LS2\_ALL\_VN\_ALL\_UNK\_UNK\_TAPX.f1bf430a-65cf-418a-b43e-28dfa9168045\_TAPU.e8811fd7-587b-4b1d-b685-d9586955086f',      '\_fbp': 'fb.1.1704294114476.1292406438',      '\_ga': 'GA1.1.1519596821.1704294091',      '\_ga\_7QN4MZMLVG': 'GS1.1.1704336969.1.1.1704336971.0.0.0',      '\_ga\_L9MPNN6QJB': 'GS1.1.1704336969.1.1.1704336980.0.0.0',      '\_ga\_S9GLR1RQFJ': 'GS1.1.1704334874.3.1.1704340301.60.0.0',      '\_ga\_W6PZ1YEX5L': 'GS1.1.1704338649.1.1.1704339309.0.0.0',      '\_gcl\_au': '1.1.737102976.1704294095',      '\_hjAbsoluteSessionInProgress': '1',      '\_hjIncludedInSessionSample\_522327': '0',      '\_hjSessionUser\_522327': 'eyJpZCI6IjBkZmM0NDMwLTNmYjctNTViMy1iYzc0LTc5ZjU1NjQ2Y2QxYyIsImNyZWF0ZWQiOjE3MDQyOTQwOTU5NDUsImV4aXN0aW5nIjp0cnVlfQ==',      '\_hjSession\_522327': 'eyJpZCI6ImEwNGM2ODJiLWZmZDEtNDU0Ni1iMzJlLTU3NTZiNWJhZjM4ZCIsImMiOjE3MDQzMzMwMTU4MzEsInMiOjAsInIiOjAsInNiIjowfQ',      '\_trackity': '7e2ddf92-8de0-c763-991f-c6e7161b2dac',      'amp\_99d374': '6gD267ETVSbykVTUmYnfSE...1hj92jli6.1hj99cat5.5s.8i.ee',      'cto\_bundle': 'LnIg-F83MXlqbUVEb01wcnVNeG5TTG41VkZLUG5vMjdobHlTM1JmTyUyQkt5aXczYTdXaXRta29ROWpVMHlzVWZtWnFuRGF5MlBNU0NMcSUyQmdtYlh5QVdTT3gzVk1pUklwSmZaN0hON1BjZnR2ZURLWDJtazQxWjVUS0hxY1p2S2lJZHlFNkxHQjUlMkY3ZUclMkZTWDliazFoekc5SWdQQSUzRCUzRA',      'delivery\_zone': 'Vk4wMzkwMDYwMDE=',      'dtdz': '-1',      'rl\_anonymous\_id': 'StackityEncrypt%3AU2FsdGVkX19IHHSuW%2Fa7YdscvoeaIiNu1AQ6XK0r6ed3lALdJO0T1fTAs8oSKVQWGif7Z0QECLkL3%2FZg56mQYw%3D%3D',      'rl\_group\_id': 'StackityEncrypt%3AU2FsdGVkX1%2Bzr5RcL6%2BpdRK%2FM6MoTfF%2BqGSXpaziCc4%3D',      'rl\_group\_trait ': 'StackityEncrypt%3AU2FsdGVkX19Raa%2FMmvCJG3SIznh%2F1j6yT4M3qjlOdPs%3D',      'rl\_page\_init\_referrer': 'StackityEncrypt%3AU2FsdGVkX1%2FU4RPIsBYghwLD8N1pAFHzbt6GJ3rv%2BP%2FM%2BBc3tcdL173VtWCXdYmo',      'rl\_page\_init\_referring\_domain': 'StackityEncrypt%3AU2FsdGVkX19LlVhPZty3IE7%2FHzN5dsBc%2BTaRpnHsj%2F3JfwNfMa%2BVgsAeIbkgEjxW',      'rl\_trait': 'StackityEncrypt%3AU2FsdGVkX1%2FMH1NKb%2Bqy2ITv0sekVVS0RYOAFe9Xr2c%3D',      'rl\_user\_id ': 'StackityEncrypt%3AU2FsdGVkX1%2Bivmh9WZr8a0rYPL0hPrKk%2BqDRivoA7uU%3D',      'tiki\_client\_id': '1519596821.1704294091'  }  headers = {      'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36',      'Accept': 'application/json, text/plain, \*/\*',      'Accept-Language': 'vi-VN,vi;q=0.9,fr-FR;q=0.8,fr;q=0.7,en-US;q=0.6,en;q=0.5',      'Referer': 'https://tiki.vn/set-2-qua-n-du-i-nam-qua-n-short-gio-nam-the-thao-basic-tre-trung-nang-do-ng-thoa-ng-ma-t-co-gia-n-4-chie-u-mrm-manlywear-p99968881.html?itm\_campaign=CTP\_YPD\_TKA\_PLA\_UNK\_ALL\_UNK\_UNK\_UNK\_UNK\_X.284641\_Y.1866961\_Z.3907942\_CN.03%2F2024---CB-2-Q%C4%90-Tron---Auto&itm\_medium=CPC&itm\_source=tiki-ads&spid=99968992',      'x-guest-token': 'YcN0C1W8zP6ElLJkTgdAMyxrpvQwb94q',      'Connection': 'keep-alive',      'TE': 'Trailers',  }  params = (      ('platform', 'web'),      ('spid', 99968992)      #('include', 'tag,images,gallery,promotions,badges,stock\_item,variants,product\_links,discount\_tag,ranks,breadcrumbs,top\_features,cta\_desktop'),  ) |

Tạo hàm lấy thông tin ở phần Preview.

|  |
| --- |
| def parser\_product(json):      d = dict()      d['id'] = json.get('id')      d['brand\_id'] = json.get('brand').get('id')      d['brand\_name'] = json.get('brand').get('name')      d['product\_name'] = json.get('name')      d['categories'] = json.get('categories').get('name')      d['discount'] = json.get('discount')      d['discount\_rate'] = json.get('discount\_rate')      d['list\_price'] = json.get('list\_price')      d['price'] = json.get('price')      d['rating\_average'] = json.get('rating\_average')      d['review\_count'] = json.get('review\_count')      d['stock\_item\_qty'] = json.get('stock\_item').get('qty')      d['stock\_item\_max\_sale\_qty'] = json.get('stock\_item').get('max\_sale\_qty')      d['short\_url'] = json.get('short\_url')      return d |

Lấy danh sách id sản phẩm để chuẩn bị cho việc crawl.

|  |
| --- |
| df\_id = pd.read\_csv('crawl\_product\_id.csv')  p\_ids = df\_id.id.to\_list()  print(p\_ids)  result = [] |

Bắt đầu crawl dữ liệu.

|  |
| --- |
| for pid in tqdm(p\_ids, total=len(p\_ids)):      response = requests.get('https://tiki.vn/api/v2/products/{}'.format(pid), headers=headers, params=params, cookies=cookies)      if response.status\_code == 200:          print('\nCrawl data {} success !!!'.format(pid))          result.append(parser\_product(response.json()))      # time.sleep(random.randrange(3, 5))  df\_product = pd.DataFrame(result) |

Lưu thông tin sản phẩm vào file csv.

|  |
| --- |
| df\_product.to\_csv('crawl\_product\_data.csv', index=False) |

## 1.2 Giới thiệu về tập dữ liệu đã crawl

Đây là tập dữ liệu về các sản phẩm thuộc danh mục thời trang nam trên trang web thương mại điện tử Tiki.vn. Dưới đây là chi tiết các trường dữ liệu đã crawl được.

* **id:** id sản phẩm.
* **brand\_id:** id của nhãn hàng.
* **brand\_name:** tên nhãn hàng.
* **product\_name:** tên sản phẩm.
* **categories:** danh mục.
* **discount:** giảm giá.
* **discount\_rate:** tỉ lệ giảm giá.
* **list\_price:** giá gốc của sản phẩm.
* **price:** giá sản phẩm sau khi giảm giá.
* **rating\_average:** đánh giá trung bình.
* **review\_count:** số lượng đánh giá.
* **stock\_item\_qty:** số lượng hàng tồn kho.
* **stock\_item\_max\_sale\_qty:** giá trị lớn nhất của số lượng sản phẩm đã bán (sale quantity) cho mỗi mặt hàng hoặc sản phẩm trong một khoảng thời gian cụ thể.
* **short\_url:** link sản phẩm.

Ảnh tổng quan về tập dữ liệu.

A screenshot of a computer screen

Description automatically generated

Hình : Tổng quan về tập dữ liệu

# CHƯƠNG 2: PHÂN TÍCH TỔNG QUAN VỀ DỮ LIỆU

## 2.1 Đọc dữ liệu

### 2.1.1 Nhập các thư viện cần thiết

Nhập các thư viện cần thiết.

|  |
| --- |
| library(dplyr)  library(ggplot2) |

### 2.1.2 Thông tin tập dữ liệu

Hiển thị thư mục làm việc.

|  |
| --- |
| getwd() |

Kết quả:



Hình : Kết quả hiển thị thư mục làm việc

Nhập tập dữ liệu.

|  |
| --- |
| # Nhập tập dữ liệu  df <- read.csv("C:/Users/Karrot/Documents/Đại học/Năm 2/Crawl Data - R/tiki\_crawldata/crawl\_product\_data.csv") |

In ra số hàng và số cột của tập dữ liệu.

|  |
| --- |
| # Trả về số hàng và số cột  dim(df) |

Kết quả:



Hình : Số hàng và số cột của tập dữ liệu

In ra tên của các trường dữ liệu.

|  |
| --- |
| # Trả về tên các cột  names(df) |

Kết quả:

A close-up of a computer code

Description automatically generated

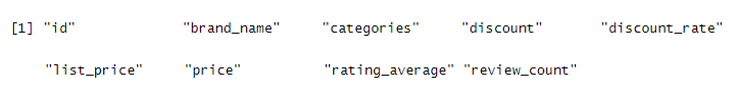
Hình : Trả về tên các trường dữ liệu

## 2.2 Tiền xử lý dữ liệu

Xóa các cột không cần thiết.

|  |
| --- |
| df <- df %>% select  (-brand\_id,-product\_name,-stock\_item\_qty,-stock\_item\_max\_sale\_qty,-short\_url)  # In kết quả  print(df) |

Kết quả:



Hình : Dataframe mới

Tìm và xóa các giá trị bị trùng lặp.

|  |
| --- |
| # Tìm các dòng bị trùng lặp trong DataFrame  row\_duplicated <- df[duplicated(df), ]  # Hiển thị DataFrame chứa các dòng bị trùng lặp  print(row\_duplicated) |

Kết quả:

A screenshot of a computer

Description automatically generated

Hình : Các dòng bị trùng lặp

Loại bỏ giá trị trùng lặp.

|  |
| --- |
| # Loại bỏ các dòng trùng lặp, giữ lại dòng đầu tiên xuất hiện  df <- unique(df)  # Hiển thị lại DataFrame không chứa dòng trùng lặp  print(df) |

Kết quả:

A screenshot of a computer

Description automatically generated

Hình : Hiển thị lại DataFrame không chứa dòng trùng lặp

Xử lý các giá trị NaN.

|  |
| --- |
| # Xử lý giá trị NaN ("Root")  df <- df[!(df$categories == "Root"), ]  df |

## 2.3 Thống kê mô tả

Thống kê mô tả tập dữ liệu.

|  |
| --- |
| summary(df) |

Kết quả:

A screenshot of a computer

Description automatically generated

Hình : Thống kê mô tả tập dữ liệu

* **id:** Cột chứa các giá trị id. Các giá trị bao gồm giá trị tối thiểu (Min.), giá trị ở phân vị 25% (1st Qu.), giá trị trung vị (Median), giá trị trung bình (Mean), giá trị ở phân vị 75% (3rd Qu.), và giá trị tối đa (Max.).
* **brand\_name:** Cột chứa thông tin về tên thương hiệu (brand\_name). Đây có vẻ là kiểu dữ liệu ký tự (character). Không có thống kê mô tả nào được hiển thị vì đây là biến phân loại.
* **categories:** Cột chứa thông tin về loại sản phẩm (categories). Cũng là kiểu dữ liệu ký tự (character) và không có thống kê mô tả.
* **discount:** Cột chứa giá trị giảm giá. Các thống kê mô tả bao gồm giá trị tối thiểu, phân vị 25%, trung vị, trung bình, phân vị 75%, và giá trị tối đa.
* **discount\_rate:** Cột chứa tỷ lệ giảm giá. Thống kê mô tả tương tự như cột discount.
* **list\_price:** Cột chứa giá niêm yết. Thống kê mô tả bao gồm giá trị tối thiểu, phân vị 25%, trung vị, trung bình, phân vị 75%, và giá trị tối đa.
* **price:** Cột chứa giá bán. Thống kê mô tả giống như cột list\_price.
* **rating\_average:** Cột chứa giá trị trung bình đánh giá. Thống kê mô tả bao gồm giá trị tối thiểu, phân vị 25%, trung vị, trung bình, phân vị 75%, và giá trị tối đa.
* **review\_count:** Cột chứa số lượng đánh giá. Thống kê mô tả bao gồm giá trị tối thiểu, phân vị 25%, trung vị, trung bình, phân vị 75%, và giá trị tối đa.

## 2.4 Trực quan hóa dữ liệu

### 2.4.1 Biểu đồ cột

Vẽ biểu đồ cột.

|  |
| --- |
| ggplot(df, aes(x = categories)) +    geom\_bar() +    labs(title = "Biểu đồ cột dọc cho categories", x = "Categories", y = "Số lượng") +    theme\_minimal() +    theme(      axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust = 1),      plot.title = element\_text(hjust = 0.5)  # Đặt title ở giữa    ) |

Kết quả:

A graph with many columns

Description automatically generated with medium confidence

Hình : Biểu đồ cột

Biểu đồ này thể hiện số lượng của từng loại trong danh mục (categories).

### 2.4.2 Biểu đồ tròn

Phân loại thành các biến thành áo, quần và khác.

|  |
| --- |
| # Lọc dữ liệu theo yêu cầu  ao <- nrow(df %>%filter(grepl("áo", categories, ignore.case = TRUE)))  quan <- nrow(df %>%filter(grepl("quần", categories, ignore.case = TRUE)))  khac <- nrow(df %>%filter(!grepl("áo", categories, ignore.case = TRUE) & !grepl("quần", categories, ignore.case = TRUE))) |

Vẽ biểu đồ tròn thể hiện tỉ lệ phân loại của categories.

|  |
| --- |
| data <- data.frame(    category = c("Áo", "Quần", "Khác"),    count = c(ao, quan, khac)  )  # Tính tổng số lượng  total\_count <- sum(data$count)  # Tính phần trăm  data$percentage <- (data$count / total\_count) \* 100  # Biểu đồ tròn thể hiện tỉ lệ và thêm phần trăm, xóa số 100, 200, 300 ngoài vòng tròn  ggplot(data, aes(x = "", y = count, fill = category)) +    geom\_bar(stat = "identity", width = 1) +    geom\_text(aes(label = sprintf("%.1f%%", percentage)), position = position\_stack(vjust = 0.5)) +    coord\_polar("y") +    labs(title = "Biểu đồ tròn tỉ lệ giữa Áo, Quần và Khác", fill = "Chú thích", x = "Tỉ lệ (%)") +    theme\_minimal() +    theme(legend.position = "bottom", axis.text = element\_blank()) |

Kết quả:

A pie chart with numbers and a red circle

Description automatically generated

Hình : Biểu đồ tròn

### 2.4.2 Biểu đồ histogram

Vẽ biểu đồ histogram dựa trên mật độ phân phối của giá sản phẩm.

|  |
| --- |
| ggplot(df, aes(x = list\_price)) +    geom\_histogram(bins = 30, fill = "lightblue", color = "black") +    labs(title = "Histogram of Price", x = "List Price", y = "Frequency") |

Kết quả:

A graph of a graph

Description automatically generated

Hình : Biểu đồ histogram

# CHƯƠNG 3: CHỨNG MINH VÀ PHÂN TÍCH CÁC GIẢ THUYẾT, BÀI TOÁN

## 3.1 Chứng minh giả thuyết (Hypothesis Testing)

Để chứng minh giả thuyết (hypothesis testing) cho một biến cụ thể trong tập dữ liệu, ta cần xác định giả thuyết null (H0) và giả thuyết chính (H1), sau đó thực hiện kiểm định thống kê để đưa ra quyết định về việc bác bỏ hay không bác bỏ giả thuyết null.

Dưới đây là một ví dụ giả định chứng minh sự khác biệt về giá trị trung bình của cột price giữa hai nhóm sản phẩm có brand\_name khác nhau (OEM và LADOS).

**Bước 1:** Xác định giả thuyết.

* Giả thuyết null (H0): Giá trị trung bình của price đối với brand\_name OEM bằng giá trị trung bình của price đối với brand\_name LADOS.
* Giả thuyết chính (H1): Giá trị trung bình của price đối với brand\_name OEM không bằng giá trị trung bình của price đối với brand\_name LADOS.

**Bước 2:** Thực hiện kiểm định t.

Trong đoạn mã bên dưới, ta giả định rằng giá trị của cột price phụ thuộc vào brand\_name. Sau đó, tôi thực hiện kiểm định t để kiểm tra xem có sự khác biệt đáng kể về giá trị trung bình của price giữa hai nhóm OEM và LADOS hay không.

|  |
| --- |
| # Kiểm định t về sự khác biệt giữa giá trị trung bình của price đối với OEM và LADOS  t\_test\_result <- t.test(df$price[df$brand\_name == "OEM"], df$price[df$brand\_name == "LADOS"])  # Hiển thị kết quả  print(t\_test\_result) |

Kết quả:

A white background with black text

Description automatically generated

Hình : Kết quả kiểm định t

* **t = -1.2622** (Giá trị thống kê t): Nếu giá trị này gần 0, có nghĩa là không có sự khác biệt đáng kể giữa hai nhóm. Trong trường hợp này, giá trị t là âm, cho biết trung bình của nhóm "OEM" có thể nhỏ hơn so với nhóm "LADOS".
* **df = 25.291** (Độ tự do của phân phối t): Đây là một ước lượng dựa trên dữ liệu và được sử dụng để xác định giá trị p.
* **p-value = 0.2184:** Đây là xác suất để thấy giá trị t thực tế hoặc lớn hơn, giả sử giả thuyết null là đúng (không có sự khác biệt giữa các nhóm). Nếu giá trị p lớn, chúng ta không thể bác bỏ giả thuyết null. Ở đây, giá trị p lớn (0.2184), nên không có đủ bằng chứng để bác bỏ giả thuyết null.
* **alternative hypothesis: true difference in means is not equal to 0** (Giả thuyết thay thế): Sự khác biệt thực sự giữa trung bình không bằng 0. Điều này có nghĩa là kiểm định xem có sự khác biệt nào đó giữa hai nhóm hay không.
* **95 percent confidence interval: -101077.17 to 24233.45** (Khoảng tin cậy 95%): Nếu giá trị 0 nằm trong khoảng này, chúng ta không thể kết luận rằng có sự khác biệt đáng kể. Trong trường hợp này, khoảng tin cậy chứa giá trị 0, cũng hỗ trợ cho việc không có sự khác biệt đáng kể.
* **sample estimates: mean of x mean of y: 169039.7 207461.5:** Ước lượng trung bình của nhóm "OEM" là 169039.7 và của nhóm "LADOS" là 207461.5.

Tóm lại, dựa vào giá trị p lớn (0.2184) và khoảng tin cậy chứa giá trị 0, không có đủ bằng chứng để bác bỏ giả thuyết null, và ta không có đủ lý do để tin rằng có sự khác biệt đáng kể giữa trung bình của hai nhóm "OEM" và "LADOS".

**Bước 3:** Kiểm tra giả thuyết.

Kết quả của kiểm định t sẽ cung cấp giá trị p (p-value). Nếu giá trị p nhỏ hơn một ngưỡng ý nghĩa (thường là 0.05), ta có thể bác bỏ giả thuyết null và chấp nhận giả thuyết chính, kết luận rằng có sự khác biệt đáng kể giữa hai nhóm.

|  |
| --- |
| # Kiểm tra giả thuyết null  if (t\_test\_result$p.value < 0.05) {    print("Bác bỏ giả thuyết null: Có sự khác biệt đáng kể giữa OEM và LADOS")  } else {    print("Chấp nhận giả thuyết null: Không có sự khác biệt đáng kể giữa OEM và LADOS")  } |

Kết quả:



Hình : Kết quả kiểm tra giả thuyết

## 3.2 Dự đoán

### 3.2.1 Linear Regression

|  |
| --- |
| # Mô hình hồi quy tuyến tính  model\_lm <- lm(price ~ discount + discount\_rate + list\_price + rating\_average, data = df)  # Hiển thị tóm tắt mô hình  summary(model\_lm) |

Kết quả:

A screenshot of a computer code

Description automatically generated

Hình : Tóm tắt mô hình Linear Regression

1. **Residuals (Dư thừa):**

* **Min:** Giá trị dư thừa nhỏ nhất.
* **1Q (Phân vị 25%):** Khoảng 25% dữ liệu có giá trị dư thừa nhỏ hơn giá trị này.
* **Median (Trung vị):** Giá trị dư thừa tại vị trí trung vị (50% dữ liệu có giá trị dư thừa nhỏ hơn).
* **3Q (Phân vị 75%):** Khoảng 75% dữ liệu có giá trị dư thừa nhỏ hơn giá trị này.
* **Max:** Giá trị dư thừa lớn nhất.

1. **Coefficients (Hệ số):**

* **Estimate:** Ước lượng của hệ số tương ứng với mỗi biến độc lập.
* **Std. Error (Độ lệch chuẩn):** Độ lệch chuẩn của ước lượng.
* **t value (Giá trị t):** Giá trị thống kê t.
* **Pr(>|t|) (Giá trị p):** Giá trị p, xác suất của giá trị t hiện tại hoặc lớn hơn nếu giả thuyết null là đúng.

1. **Residual standard error (Sai số tiêu chuẩn của dư thừa):** Độ lệch chuẩn của giá trị dư thừa, đo lường sự biến động của các điểm dữ liệu quanh đường hồi quy.
2. **Multiple R-squared (R-quadrat đa biến):** Đo lường mức độ giải thích bởi mô hình. Giá trị 1 có nghĩa là mô hình giải thích toàn bộ biến động của phản hồi.
3. **Adjusted R-squared (R-quadrat được điều chỉnh):** Đo lường mức độ giải thích điều chỉnh cho số lượng biến động trong mô hình.
4. **F-statistic (Thống kê F):** Kiểm định F cho biết xem có sự khác biệt đáng kể giữa mô hình được fit và mô hình không có biến động nào không.
5. **p-value (Giá trị p):** Xác suất để thấy giá trị F hiện tại hoặc lớn hơn nếu giả thuyết null là đúng.

Tóm lại:

* Mô hình có vẻ rất phù hợp với dữ liệu, với R-quadrat đa biến và R-quadrat được điều chỉnh đều bằng 1.
* Các hệ số của các biến như "discount", "discount\_rate", "list\_price", "rating\_average" được ước lượng.
* Các giá trị p cho tất cả các hệ số đều rất nhỏ, đặc biệt là cho "discount", "list\_price" (p-value < 2.2e-16), có nghĩa là chúng có ảnh hưởng đáng kể đến biến phụ thuộc "price".
* F-statistic là một giá trị rất lớn và giá trị p của nó cũng rất nhỏ, có nghĩa là mô hình là đủ tốt để giải thích dữ liệu.

Vẽ biểu đồ trực quan.

|  |
| --- |
| # Vẽ biểu đồ scatterplot với đường hồi quy  library(ggplot2)  ggplot(df, aes(x = list\_price, y = price)) +    geom\_point(color = "blue", size = 3) +    geom\_smooth(method = "lm", se = FALSE, color = "red", formula = y ~ x) +    labs(title = "Linear Regression", x = "List Price", y = "Price") +    theme\_minimal() |

Kết quả:

A graph with a line and dots

Description automatically generated with medium confidence

Hình : Biểu đồ Linear Regression

### 3.2.2 K-Means

Phân loại dữ liệu thành 3 cụm dựa trên các biến "discount", "discount\_rate", "list\_price", và "rating\_average".

|  |
| --- |
| # K-Means Clustering  model\_kmeans <- kmeans(df[, c("discount", "discount\_rate", "list\_price", "rating\_average")], centers = 3)  # Thêm thông tin phân cụm vào dữ liệu  df$cluster <- as.factor(model\_kmeans$cluster) |

Kết quả:

A grid of black and white squares

Description automatically generated

Hình : Thông tin phân cụm dữ liệu

1. **Cluster means (Trung tâm cụm):**
   1. Cluster 1:
      1. discount: 324,500.00
      2. discount\_rate: 35.26
      3. list\_price: 1,029,526.3
      4. rating\_average: 3.00
   2. Cluster 2:
      1. discount: 110,310.00
      2. discount\_rate: 26.43
      3. list\_price: 415,719.6
      4. rating\_average: 4.13
   3. Cluster 3:
      1. discount: 22,624.79
      2. discount\_rate: 12.70
      3. list\_price: 150,693.1
      4. rating\_average: 4.13
2. **Clustering vector (Vector phân cụm):**
   1. Liệt kê cụm mà mỗi quan sát thuộc về. Ví dụ, quan sát thứ 1 thuộc về cụm 2, quan sát thứ 2 thuộc về cụm 3, và tiếp tục.
3. **Within cluster sum of squares by cluster (Tổng bình phương trong cụm):**
   1. Cho biết tổng bình phương của khoảng cách từ mỗi điểm đến trung tâm của cụm nó thuộc về. Giảm giá trị này là mục tiêu của việc cải thiện mô hình.
4. **Percentage of variance explained (Tỉ lệ phương sai được giải thích):**
   1. Tỉ lệ phương sai được giải thích bởi phương sai giữa các cụm so với tổng phương sai.

Trong trường hợp này, dữ liệu đã được phân thành 3 cụm, và phương sai giữa các cụm là khoảng 72.8% của tổng phương sai. Kết quả này cung cấp cái nhìn tổng quan về cách dữ liệu đã được phân loại và cụm trung tâm của từng nhóm.

Vẽ biểu đồ.

|  |
| --- |
| # Vẽ biểu đồ scatterplot  library(ggplot2)  ggplot(df, aes(x = discount, y = discount\_rate, color = cluster)) +    geom\_point() +    labs(title = "K-Means Clustering", x = "Discount", y = "Discount Rate") +    scale\_color\_manual(values = c("blue", "green", "red")) |

Kết quả:

A graph with colored dots

Description automatically generated

Hình : Biểu đồ KMeans

## 3.3 Phân tích PCA

Phân tích thành phần chính (PCA) là một phương pháp giảm chiều dữ liệu, giúp tìm ra các biến quan trọng nhất và giảm số lượng biến để dễ dàng hiểu và trực quan hóa dữ liệu. Dưới đây là một hướng dẫn sơ bộ về cách thực hiện PCA dữ liệu.

**Bước 1:** Chuẩn bị dữ liệu.

|  |
| --- |
| # Lựa chọn các biến số cần tham gia vào PCA  df\_pca <- df[, c("discount", "list\_price", "price")]  # Chuẩn bị dữ liệu cho PCA (loại bỏ các cột không phải dạng số và xử lý missing data)  df\_pca <- na.omit(df\_pca)  df\_pca\_scaled <- scale(df\_pca)  # Chuẩn hóa dữ liệu |

**Bước 2:** Thực hiện PCA.

|  |
| --- |
| # Thực hiện PCA  pca\_result <- prcomp(df\_pca\_scaled)  # Hiển thị kết quả  print(summary(pca\_result)) |

Kết quả:

A black text on a white background

Description automatically generated

Hình : Kết quả PCA

Bước 3: Trực quan hóa kết quả PCA.

|  |
| --- |
| # Biểu đồ các thành phần chính  biplot(pca\_result, scale = 0) |

Kết quả:

A graph of a price

Description automatically generated with medium confidence

Hình : Biểu đồ trực quan PCA

# LINK MÃ NGUỒN

Link Github: [Tiki\_CrawlData/code\_crawl\_data at main · minhkhanh-coder/Tiki\_CrawlData (github.com)](https://github.com/minhkhanh-coder/Tiki_CrawlData/tree/main/code_crawl_data)

# TÀI LIỆU THAM KHẢO

# KẾT QUẢ KIỂM TRA ĐẠO VĂN

**ĐẠI HỌC HUẾ**

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

🙢

**PHIẾU ĐÁNH GIÁ ĐỒ ÁN/TIỂU LUẬN/BÀI TẬP LỚN**

**Học kỳ I, năm học 2021 – 2022**

|  |  |
| --- | --- |
| **Cán bộ chấm thi 1** | **Cán bộ chấm thi 2** |
| **Nhận xét:**  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  Điểm đánh giá của CBCT1:  Bằng số: ..............................................  Bằng chữ: ............................................ | **Nhận xét:**  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  Điểm đánh giá của CBCT2:  Bằng số: ..............................................  Bằng chữ: ............................................ |

Điểm kết luận: ..................................................................................................................

Bằng số: ............................................................................................................................

Bằng chữ: ..........................................................................................................................

*Thừa Thiên Huế, ngày 15 tháng 01 năm 2022*

**Cán bộ chấm thi 1**

*(Ký và ghi rõ họ tên)*

**Cán bộ chấm thi 2**

*(Ký và ghi rõ họ tên)*