ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

A picture containing text, vector graphics

Description automatically generated

**BÁO CÁO ĐỒ ÁN**

**DỮ LIỆU LỚN**

**Đề tài:**

**PHÂN TÍCH BỘ DỮ LIỆU VỀ SỰ TƯƠNG TÁC**

**CÁC BÀI ĐĂNG BÁN HÀNG TRÊN FACEBOOK LIVE TẠI THÁI LAN**

Giảng Viên Hướng Dẫn: **ThS. Nguyễn Hồ Duy Trí**

Lớp: **IS405.O11**

Nhóm: **10**

**Trần Gia Phong**  **20521748**

**Nguyễn Hải Đăng**  **20521158**

**Dương Ngọc Hải**  **20521275**

**Nguyễn Tiến Nhân**  **20521702**

Thành phố Hồ Chí Minh, ngày 2 tháng 6 năm 2024

# NHẬN XÉT CỦA GIÁO VIÊN

**LỜI CẢM ƠN**

Trước tiên, chúng em xin bày tỏ lòng biết ơn sâu sắc và chân thành nhất đến Ban Giám hiệu Trường Đại học Công Nghệ Thông Tin - Đại học Quốc gia Thành phố Hồ Chí Minh và Khoa Hệ Thống Thông Tin vì đã tạo điều kiện hỗ trợ và giúp đỡ chúng em trong suốt quá trình học tập và thực hiện đồ án môn học này.

Tiếp theo, chúng em xin gửi lời cảm ơn chân thành tới giảng viên bộ môn Dữ liệu Lớn, Thạc sĩ Nguyễn Hồ Duy Trí. Chúng em rất biết ơn Thầy đã truyền đạt những vốn kiến thức quý báu, tận tình hướng dẫn, quan tâm và động viên chúng em trong suốt quá trình thực hiện đề tài. Những tài liệu và kinh nghiệm mà Thầy chia sẻ đã là hành trang vô cùng quý giá cho chúng em.

Cuối cùng, chúng em xin gửi lời cảm ơn chân thành đến tất cả các bạn trong nhóm. Cảm ơn các bạn đã cùng nhau chia sẻ công việc, hoàn thành tốt trách nhiệm của cá nhân dưới sự hướng dẫn của Thầy và sự phân công của nhóm trưởng. Các bạn là những nhân tố quan trọng không thể thiếu, là chìa khóa để hoàn thành đề tài.

Mặc dù đã cố gắng hoàn thành đề tài với tất cả nỗ lực, nhưng chúng em vẫn mong nhận được sự thông cảm và những đóng góp, nhận xét quý báu từ Thầy. Những lời góp ý từ Thầy sẽ là hành trang vô cùng quý giá để chúng em vận dụng cho những môn học khác trong tương lai.

Chúng em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, tháng năm

Nhóm sinh viên thực hiện

# MỤC LỤC

[NHẬN XÉT CỦA GIÁO VIÊN 2](#_Toc168246258)

[MỤC LỤC 4](#_Toc168246259)

[DANH MỤC BẢNG 7](#_Toc168246260)

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 8](#_Toc168246261)

[1.1 Lý do chọn đề tài 8](#_Toc168246262)

[1.2 Dữ liệu 8](#_Toc168246263)

[**1.2.1 Bộ dữ liệu** 8](#_Toc168246264)

[**1.2.2 Mô tả dữ liệu** 9](#_Toc168246265)

[**1.2.3 Thống kê dữ liệu** 10](#_Toc168246266)

[1.2 Mô tả bài toán 16](#_Toc168246267)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 17](#_Toc168246268)

[2.1 Kỹ thuật tiền xử lý dữ liệu 17](#_Toc168246269)

[2.1.1 Tải và Kiểm Tra Dữ Liệu (Data Loading and Inspection) 17](#_Toc168246270)

[2.1.2 Chuyển Đổi Kiểu Dữ Liệu (Data Type Conversion) 17](#_Toc168246271)

[2.1.3 Loại Bỏ Cột (Dropping Columns) 18](#_Toc168246272)

[2.1.4 Mã Hóa Nhãn (Label Encoding) 18](#_Toc168246273)

[2.2 Các thuật toán áp dụng 19](#_Toc168246274)

[2.2.1 Linear Regression 19](#_Toc168246275)

[2.2.1 K-Means 20](#_Toc168246276)

[2.3 Phương pháp đánh giá thuật toán 21](#_Toc168246277)

[2.3.1 R-Squared (R²): 21](#_Toc168246278)

[2.3.2 Root Mean Square Deviation (RMSD): 22](#_Toc168246279)

[2.3.3 So sánh giá trị dự đoán với giá trị thực tế: 22](#_Toc168246280)

[CHƯƠNG 3: TRIỂN KHAI MÔ HÌNH 24](#_Toc168246281)

[3.1 Tiền xử lý dữ liệu 24](#_Toc168246282)

[3.2 Triển khai mô hình 33](#_Toc168246283)

[CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC 36](#_Toc168246284)

[4.1 Kết quả chạy thuật toán 36](#_Toc168246285)

[4.2 Đánh giá 37](#_Toc168246286)

[CHƯƠNG 5: KẾT LUẬN 40](#_Toc168246287)

[5.1 Ưu điểm và hạn chế 40](#_Toc168246288)

[5.1.1 Ưu điểm 40](#_Toc168246289)

[5.1.2 Hạn chế 40](#_Toc168246290)

[5.2 Hướng phát triển 40](#_Toc168246291)

[PHÂN CÔNG CÔNG VIỆC 41](#_Toc168246292)

[TÀI LIỆU THAM KHẢO 42](#_Toc168246293)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Minh họa bộ dữ liệu Facebook Live sellers in Thailand, UCI ML Rep 10](#_Toc168246294)

[Hình 2. Các giá trị min, max, mean, median của các cột giá trị số 12](#_Toc168246295)

[Hình 3. Các giá trị phổ biến và ít phổ biến trên các cột dữ liệu kiểu số 13](#_Toc168246296)

[Hình 4. Các giá trị phổ biến nhất và ít phổ biến nhất trên cột giờ 13](#_Toc168246297)

[Hình 5. Biểu đồ so sánh số lượng bài viết theo từng loại trạng thái 14](#_Toc168246298)

[Hình 6. Biểu đồ stack các loại phản ứng trên từng loại bài đăng 14](#_Toc168246299)

[Hình 7. Biểu đồ stack các loại tương tác trên từng loại bài viết 15](#_Toc168246300)

[Hình 8. Ma trận tương quan giữa các thuộc tính 16](#_Toc168246301)

[Hình 9. Code Ví dụ để tải và kiểm tra dữ liệu từ tệp csv 18](#_Toc168246302)

[Hình 10. Code ví dụ về cách chuyển đổi kiểu dữ liệu từ chuỗi sang số nguyên 18](#_Toc168246303)

[Hình 11. Code loại bỏ một cột không cần thiết 19](#_Toc168246304)

[Hình 12. Code ví dụ mã hóa thuộc tính kiểu string 19](#_Toc168246305)

[Hình 13. Kết quả mã hóa 19](#_Toc168246306)

[Hình 14. Hồi quy tuyến tính 20](#_Toc168246307)

[Hình 15. Minh họa hồi quy tuyến tính 21](#_Toc168246308)

[Hình 16. Công thức tính khoảng cách Euclidean 22](#_Toc168246309)

[Hình 17. Công thức tính R² 22](#_Toc168246310)

[Hình 18. Công thức tính RMSD 23](#_Toc168246311)

[Hình 19. Ví dụ một hàm cal\_accuracy 24](#_Toc168246312)

[Hình 20. Cài đặt pySpark 25](#_Toc168246313)

[Hình 21. Khởi tạo spark session 25](#_Toc168246314)

[Hình 22. Đọc dữ liệu từ file csv 25](#_Toc168246315)

[Hình 23. Kiểm tra dữ liệu 26](#_Toc168246316)

[Hình 24. Kiểm tra số dòng dữ liệu 26](#_Toc168246317)

[Hình 25. Kiểm tra cấu trúc dữ liệu 27](#_Toc168246318)

[Hình 26. Kiểm tra có tồn tại dữ liệu null không bằng cách đếm dữ liệu không null 27](#_Toc168246319)

[Hình 27. Kiểm tra số cột dữ liệu 27](#_Toc168246320)

[Hình 28. Xóa các cột dữ liệu thừa 28](#_Toc168246321)

[Hình 29. Kiểm tra số cột sau khi xóa 28](#_Toc168246322)

[Hình 30. Kiểm tra các giá trị duy nhất 28](#_Toc168246323)

[Hình 31. Lấy dữ liệu của cột “status\_published” và chuyển thành danh sách kiểu Python 29](#_Toc168246324)

[Hình 32. Tách ngày và giờ 29](#_Toc168246325)

[Hình 33. Định dạng lại kiểu dữ liệu ngày 29](#_Toc168246326)

[Hình 34. Import các thư viện và khởi tạo hàm cần thiết 30](#_Toc168246327)

[Hình 35. Tạo một cột mới chứa giờ 30](#_Toc168246328)

[Hình 36. Lưu giờ vào một list 30](#_Toc168246329)

[Hình 37. Tạo cột chứa dữ liệu giờ 31](#_Toc168246330)

[Hình 38. Chuyển kiểu dữ liệu cột “Hour” thành integer 32](#_Toc168246331)

[Hình 39. Kiểm tra lại cấu trúc của bộ dữ liệu 32](#_Toc168246332)

[Hình 40. Encoding 33](#_Toc168246333)

[Hình 41. Triển khai trên cột “status\_type” 33](#_Toc168246334)

[Hình 42. Dữ liệu sau khi tiền xử lý 34](#_Toc168246335)

[Hình 43. Định nghĩa hàm math\_cal 34](#_Toc168246336)

[Hình 44. Chạy thuật toán Linear Regression 35](#_Toc168246337)

[Hình 45. Chia train test và ứng dụng mô hình hồi quy tuyến tính 36](#_Toc168246338)

[Hình 46. Định nghĩa hàm predict 37](#_Toc168246339)

[Hình 47. Dự đoán trên tập test 37](#_Toc168246340)

[Hình 48. Đổi kiểu dữ liệu 38](#_Toc168246341)

[Hình 49. Tạo cột mới chứa kết quả dự đoán 38](#_Toc168246342)

[Hình 50. Hàm khởi tạo R Square 39](#_Toc168246343)

[Hình 51. Hàm khởi tạo RMSD 39](#_Toc168246344)

[Hình 52. Khởi tạo hàm cal\_accuracy 39](#_Toc168246345)

[Hình 53. Dữ liệu gốc và dữ liệu dự đoán 40](#_Toc168246346)

[Hình 54. Tính R Square và RMSD 40](#_Toc168246347)

[Hình 55. Tính accuracy 40](#_Toc168246348)

# DANH MỤC BẢNG

[Bảng 1. Mô tả dữ liệu 11](#_Toc168246369)

[Bảng 2. Phân công công việc 42](#_Toc168246370)

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## Lý do chọn đề tài

Bán hàng trên mạng xã hội đã trở thành một phương thức kinh doanh vô cùng phổ biến trong những năm gần đây. Các nền tảng như Facebook, Instagram, TikTok và nhiều mạng xã hội khác cung cấp cơ hội tiếp cận với hàng triệu người dùng, mở ra những khả năng mới cho các doanh nghiệp trong việc tiếp thị và bán hàng. Một trong những lợi ích chính của bán hàng trên mạng xã hội là khả năng tiếp cận được với một nhóm khách hàng tiềm năng rộng lớn. Các công cụ quảng cáo và phân segmentation tinh vi trên các nền tảng này giúp doanh nghiệp xác định và nhắm mục tiêu đến đúng nhóm khách hàng mà họ muốn tiếp cận, từ đó tăng hiệu quả của các chiến dịch marketing. Hơn nữa, mạng xã hội còn mang lại cơ hội tương tác trực tiếp và xây dựng mối quan hệ gắn bó với khách hàng. Thông qua các tính năng như bình luận, chia sẻ và nhắn tin, doanh nghiệp có thể lắng nghe phản hồi từ khách hàng, hiểu rõ nhu cầu và mong muốn của họ, qua đó cải thiện chất lượng sản phẩm và dịch vụ.

Trong số các nền tảng mạng xã hội, Facebook được xem là một trong những kênh bán hàng trực tuyến hữu hiệu nhất. Với hàng tỷ người dùng trên toàn cầu, Facebook cung cấp nhiều tính năng như Trang Facebook, Nhóm và Marketplace để doanh nghiệp có thể quảng bá, tiếp thị và thực hiện các giao dịch trực tiếp. Các công cụ quảng cáo nâng cao trên Facebook còn giúp doanh nghiệp tối ưu hóa chiến lược tiếp cận khách hàng.

Sử dụng các giải thuật big data để phân tích dữ liệu liên quan đến bán hàng trên Facebook là một chiến lược vô cùng quan trọng để tối ưu hóa hiệu quả bán hàng trên nền tảng này. Thông qua việc thu thập và phân tích các dữ liệu lớn từ Facebook, như dữ liệu về hành vi người dùng, thông tin nhân khẩu học, tương tác với nội dung và quảng cáo, doanh nghiệp có thể tạo ra những hiểu biết sâu sắc về khách hàng của mình. Các công cụ phân tích dữ liệu tiên tiến sẽ giúp doanh nghiệp phân đoạn khách hàng một cách chính xác, xác định các phân khúc quan trọng và đưa ra các chiến lược tiếp thị phù hợp. Với sự kết hợp giữa big data và các chiến lược bán hàng trên mạng xã hội, doanh nghiệp sẽ có thể đạt được những kết quả kinh doanh ấn tượng trên Facebook.

## Dữ liệu

**1.2.1 Bộ dữ liệu**

Tên: “Facebook Live sellers in Thailand, UCI ML Repo”

Nguồn: [Facebook Live sellers in Thailand, UCI ML Repo (kaggle.com)](https://www.kaggle.com/datasets/ashishg21/facebook-live-sellers-in-thailand-uci-ml-repo?fbclid=IwZXh0bgNhZW0CMTAAAR3rGRe8OzfjuYEuJ9-6lStJmuMj5_UWg6D54IGDPdktTGemzCklAxOnuuQ_aem_AYj-Z3l0Lg9uuXgHPvkYQTSQQ9rSqfnc7jMisQxy9X5GHTDr0P7ogxbsRu_2QMPhHAf1sYcXYNyyv362YdtucRF6)

**1.2.2 Mô tả dữ liệu**

Dataset theo dõi các post được đăng trên ứng dụng Facebook, các dữ liệu như thời điểm đăng bài, loại bài viết, lượt tương tác, bình luận, chia sẻ, cảm xúc.

A screenshot of a computer

Description automatically generated

Hình 1. Minh họa bộ dữ liệu Facebook Live sellers in Thailand, UCI ML Rep

Dataset là file csv tên là Live.csv, gồm dữ liệu có 16 thuộc tính và 7050 dòng, với mô tả thuộc tính trong bảng sau:

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả thuộc tính** |
| status\_id | interger | Mã định danh của post |
| status\_type | string | Loại post |
| status\_published | string | Thời điểm người dùng đăng bài (%m-%d-%y %h-%m) |
| num\_reactions | integer | Số lượng cảm xúc của post |
| num\_comments | integer | Số lượng bình luận của post |
| num\_shares | integer | Số lượng chia sẻ của post |
| num\_likes | integer | Số lượng thích của post |
| num\_loves | integer | Số lượng yêu thích của post |
| num\_wows | integer | Số lượng wows của post |
| num\_hahas | integer | Số lượng hahas của post |
| num\_sads | integer | Số lượng buồn của post |
| num\_angrys | integer | Số lượng phẫn nộ của post |
| Column1 | string |  |
| Column2 | string |  |
| Column3 | string |  |
| Column4 | string |  |

Bảng 1. Mô tả dữ liệu

**1.2.3 Thống kê dữ liệu**

Sử dụng PySpark để trực quan hóa và nhiều tác vụ khác đối với bộ dữ liệu từ đó có thể hiểu rõ hơn để thực hiện phân tích.

Các giá trị Min, Max, mean, median:

A screenshot of a computer

Description automatically generated

Hình 2. Các giá trị min, max, mean, median của các cột giá trị số

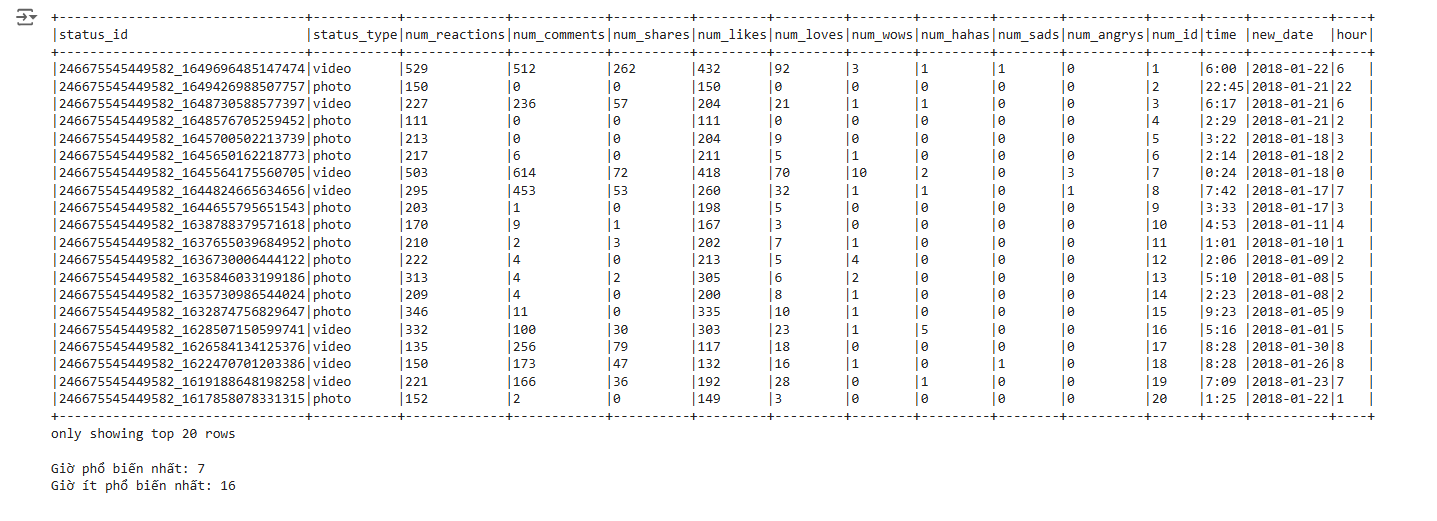
Các giá trị phổ biến nhất (mode) và ít phổ biến ̣(least frequent) nhất của các cột kiểu số:

A screenshot of a computer code

Description automatically generated

Hình 3. Các giá trị phổ biến và ít phổ biến trên các cột dữ liệu kiểu số

Các giá trị phổ biến nhất (mode) và ít phổ biến ̣(least frequent) nhất của cột giờ (sau khi tách giờ ra khỏi ngày):



Hình 4. Các giá trị phổ biến nhất và ít phổ biến nhất trên cột giờ

Khám phá các thuộc tính

Biểu đồ trực quan hóa số lượng bài viết theo từng loại trạng thái

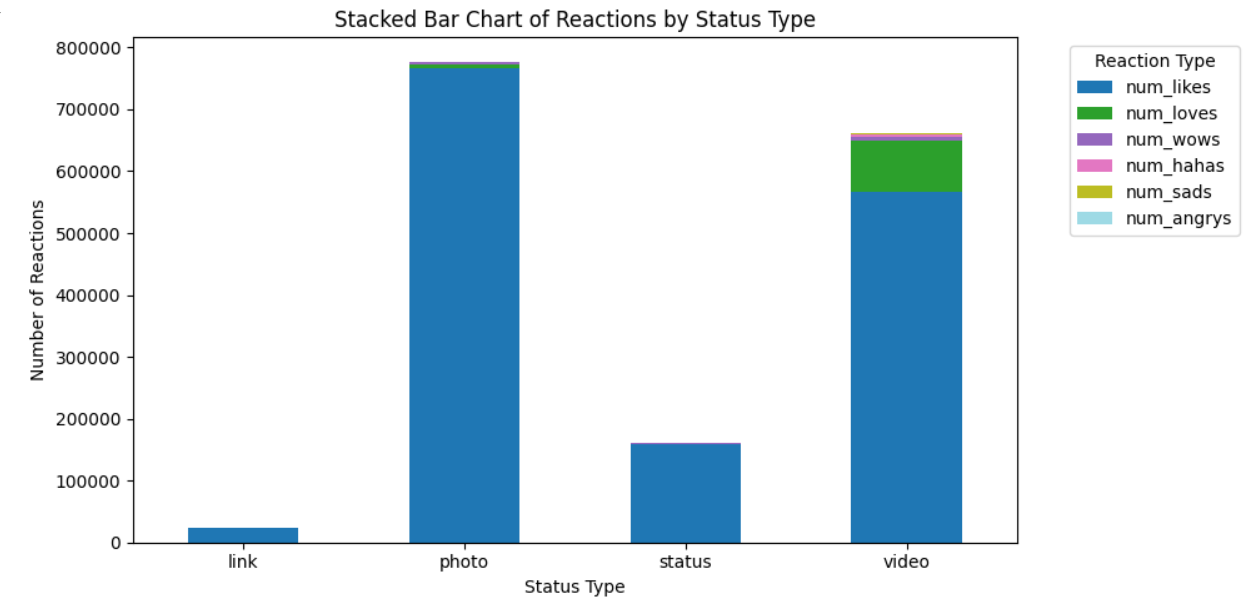
A graph with blue rectangles

Description automatically generated

Hình 5. Biểu đồ so sánh số lượng bài viết theo từng loại trạng thái

Nhận xét: Các bài viết loại video, hình ảnh chiếm số lượng lớn, trong khi loại status thuần (văn bản) và đường dẫn chiếm số lượng rất nhỏ. Điều đó cho thấy người bán thường lựa chọn thêm ảnh và video để thu hút người mua.

Biểu đồ stack các loại phản ứng (likes, loves, wows, hahas, sads, angrys) để đánh giá mức độ tương tác trên từng loại bài đăng



Hình 6. Biểu đồ stack các loại phản ứng trên từng loại bài đăng

Nhận xét: lượt likes chiếm đa số trên tất cả loại bài đăng, tuy nhiên đối với loại video, lượt yêu thích chiếm tỉ lệ lớn hơn so với trên các loại bài đăng khác. Điều này có thể ảnh hưởng đến quyết định lựa chọn loại bài đăng của người bán.

Biểu đồ stack các loại tương tác (cảm xúc, bình luận, chia sẻ) trên từng loại bài viết:

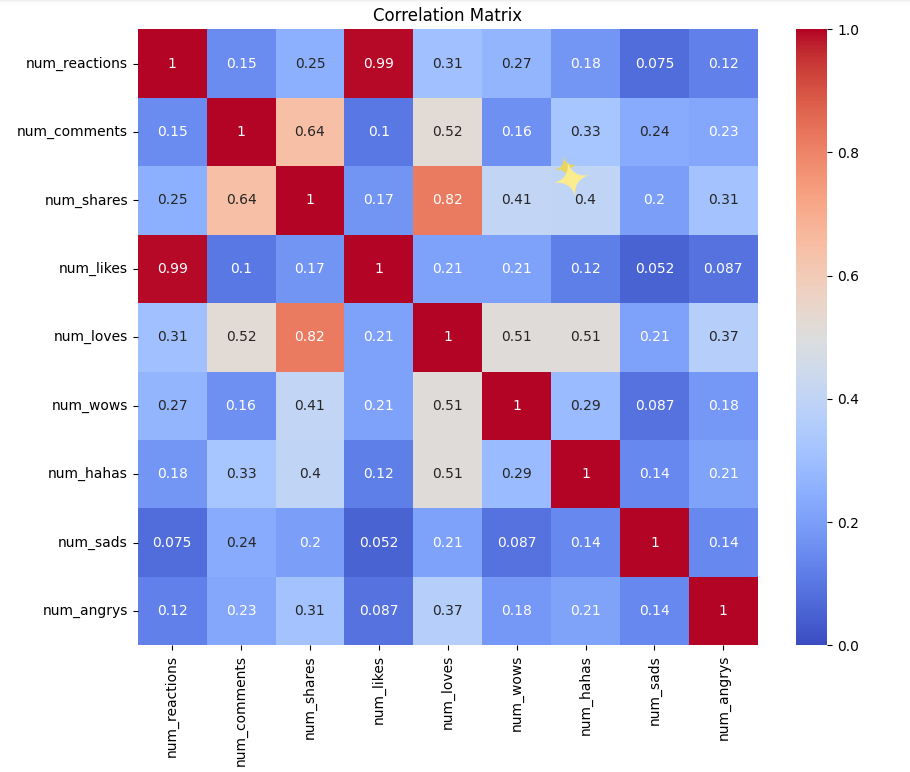
**A graph of different colored bars

Description automatically generated**

Hình 7. Biểu đồ stack các loại tương tác trên từng loại bài viết

Nhận xét: ngoại trừ loại bài đăng là video có tỉ lệ bình luận rất cao, lượt chia sẻ cũng không ít thì đặc điểm chung của các loại bài đăng còn lại là lượt tương tác cao, còn lượt bình luận và chia sẻ thấp. điều này cho thấy mức độ tương tác có hiệu quả hơn khi đăng bài bán hang có video.

Vẽ ma trận tương quan giữa các thuộc tính:



Hình 8. Ma trận tương quan giữa các thuộc tính

Nhận xét:

* “num\_reaction” và “num\_likes” có độ tương quan rất cao. Điều này cho thấy sự gia tăng đồng điệu mạnh mẽ của hai cột này, cụ thể lượt reactions cao dẫn đến lượt likes sẽ cao, độ tương quan gần như bằng 1 (0.99) cho thấy lượt likes cũng chiếm tỉ trọng cao nhất trong tổng số reactions.
* num\_share” và “num\_comments” có độ tương quan cao với nhau và cả hai đều tương quan khá cao với loves. Điều này cho thấy mức độ yêu thích bài viết có ảnh hưởng lớn đến việc bình luận và chia sẻ bài viết.
* “num\_wows” và “num\_hahas” có độ tương quan khá cao với “num\_loves”. Điều này chứng tỏ mức độ tương quan cao giữa các cảm xúc tích cực.
* “num\_sads” và “num\_angrys” có độ tương quan rất thấp với các cột khác. Từ đó cho thấy các cảm xúc tiêu cực tăng thì các lượt tương tác, phản ứng tích cực sẽ có xu hướng giảm.

## Mô tả bài toán

Để hiểu rõ hơn về hoạt động của các nhà bán hàng trực tiếp trên Facebook, chúng ta có thể khai thác và phân tích bộ dữ liệu lớn "Facebook Live sellers in Thailand, UCI ML Repo" được chia sẻ trên nền tảng Kaggle. Bộ dữ liệu này chứa thông tin chi tiết về các nhà bán hàng, như hồ sơ, dữ liệu bán hàng, các chỉ số tương tác của người dùng và nhiều đặc điểm khác.

Thông qua phân tích các thông tin về lượt tương tác, bình luận, chia sẻ, cảm xúc, giờ đăng bài của các bài đăng trên bộ dữ liệu này, nhóm sẽ dự đoán loại bài đăng của các nhà bán hàng trên Facebook Live tại Thái Lan. Điều này sẽ giúp chúng ta đưa ra các chiến lược kinh doanh hiệu quả hơn, cũng như hiểu rõ hơn về tiềm năng và thách thức của thị trường bán hàng trực tuyến thông qua nền tảng mạng xã hội ở Thái Lan.

Đề tài được thực hiện trên bộ dữ liệu Real-time Facebook Live sellers, phân tích và dự đoán loại bài đăng dựa trên tương tác của bài đăng.

* Input: giờ đăng bài, lượt bình luận, chia sẻ, tương tác, các loại cảm xúc
* Output: loại bài đăng

Cách triển khai bài toán:

* Thu thập dữ liệu trên Kaggle, sử dụng bộ dữ liệu Real-time Facebook Live sellers
* Tiền xử lý dữ liệu
* Ứng dụng giải thuật huấn luyện, xây dựng mô hình

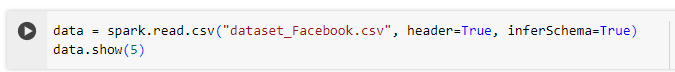
# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Kỹ thuật tiền xử lý dữ liệu

### 2.1.1 Tải và Kiểm Tra Dữ Liệu (Data Loading and Inspection)

Tải dữ liệu từ các nguồn bên ngoài là bước đầu tiên trong quá trình xử lý dữ liệu. Các nguồn dữ liệu có thể bao gồm tệp CSV, cơ sở dữ liệu, API, v.v. Sau khi dữ liệu được tải, việc kiểm tra dữ liệu giúp hiểu rõ cấu trúc, các loại dữ liệu và phát hiện các giá trị thiếu hoặc không hợp lệ.

Ví dụ để tải và kiểm tra dữ liệu từ tệp **csv** có tên “*dataset\_Facebook*” ta có thể sử dụng đoạn code sau:

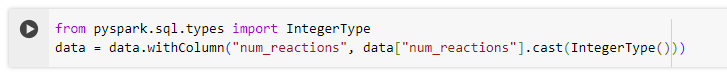


Hình 9. Code Ví dụ để tải và kiểm tra dữ liệu từ tệp csv

### 2.1.2 Chuyển Đổi Kiểu Dữ Liệu (Data Type Conversion)

Chuyển đổi kiểu dữ liệu là quá trình biến đổi các giá trị của một cột trong dữ liệu từ một kiểu dữ liệu sang kiểu dữ liệu khác. Điều này thường được thực hiện để chuẩn bị dữ liệu cho các phân tích hoặc xử lý tiếp theo, hoặc để đảm bảo tính chính xác của dữ liệu. Trong nhiều trường hợp, các thuật toán và phép toán yêu cầu dữ liệu có kiểu dữ liệu cụ thể.

Trong Python, khi làm việc với dữ liệu trong PySpark DataFrame, ta có thể cần chuyển đổi kiểu dữ liệu của các cột sang kiểu dữ liệu mong muốn. Dưới đây là một ví dụ về cách chuyển đổi kiểu dữ liệu từ chuỗi sang số nguyên:



Hình 10. Code ví dụ về cách chuyển đổi kiểu dữ liệu từ chuỗi sang số nguyên

Trong ví dụ trên, **cast()** được sử dụng để chuyển đổi kiểu dữ liệu của cột 'num\_reactions' từ chuỗi sang số nguyên. Điều này giúp làm cho dữ liệu trở nên dễ dàng thao tác và phân tích hơn.

### 2.1.3 Loại Bỏ Cột (Dropping Columns)

Loại bỏ cột là quá trình loại bỏ các cột không cần thiết hoặc không mong muốn khỏi dữ liệu. Điều này có thể cần thiết để làm cho dữ liệu gọn gàng hơn, giảm kích thước của DataFrame, hoặc tập trung vào các cột quan trọng hơn cho phân tích.

Trong một số trường hợp, khi làm việc với dữ liệu, ta có thể muốn loại bỏ một hoặc nhiều cột không cần thiết.



Hình 11. Code loại bỏ một cột không cần thiết

Trong ví dụ trên, **drop()** được sử dụng để loại bỏ cột có tên là “*unwanted\_column*” khỏi DataFrame. Điều này giúp làm cho dữ liệu trở nên gọn gàng hơn và tập trung vào các cột quan trọng hơn cho phân tích hoặc xử lý tiếp theo.

### 2.1.4 Mã Hóa Nhãn (Label Encoding)

Mã hóa nhãn là kỹ thuật chuyển đổi các giá trị của biến hạng mục thành các số nguyên, giúp các thuật toán máy học có thể xử lý chúng một cách hiệu quả hơn. Các thuật toán máy học thường làm việc tốt hơn với dữ liệu số, do đó, việc chuyển đổi các giá trị hạng mục thành số là bước quan trọng trong quá trình tiền xử lý dữ liệu.

Giả sử ta có một danh sách các giá trị hạng mục như giới tính (**'Male'** và **'Female'**) và muốn chuyển đổi chúng thành các giá trị *số*. Ta có thể thực hiện mã hóa bằng cách:

A white card with black text

Description automatically generated

Hình 12. Code ví dụ mã hóa thuộc tính kiểu string

Kết quả thu được sẽ là:

A screen shot of a computer code

Description automatically generated

Hình 13. Kết quả mã hóa

Như vậy **genders** đã được chuyển đổi thành một mảng số nguyên.Việc này giúp đơn giản hóa quá trình xử lý dữ liệu và làm cho các thuật toán máy học có thể hoạt động hiệu quả hơn với dữ liệu số.

## 2.2 Các thuật toán áp dụng

### 2.2.1 Linear Regression

Hồi quy tuyến tính (Linear Regression) là một trong những phương pháp quan trọng nhất trong học máy và thống kê, được sử dụng rộng rãi để dự đoán giá trị của một biến phụ thuộc dựa trên một hoặc nhiều biến độc lập. Thuật ngữ "**tuyến tính**" xuất phát từ việc mối quan hệ giữa các biến được mô tả bằng một đường thẳng. Trong báo cáo này, chúng em sẽ khám phá lý thuyết đằng sau hồi quy tuyến tính và cách nó được áp dụng trong PySpark.

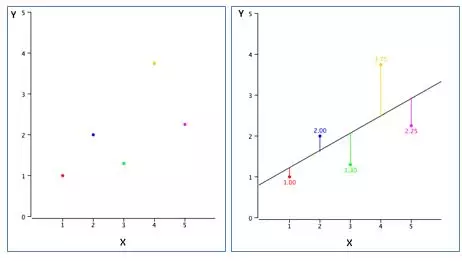


Hình 14. Hồi quy tuyến tính

Mô hình hồi quy tuyến tính giả định rằng có một mối quan hệ tuyến tính giữa biến phụ thuộc 𝑦 và các biến độc lập 𝑋. Cụ thể, mô hình được biểu diễn dưới dạng:



Trong khi sử dụng hồi quy tuyến tính, mục tiêu của chúng ta là để làm sao một đường thẳng có thể tạo được sự phân bố gần nhất với hầu hết các điểm. Do đó làm giảm khoảng cách (sai số) của các điểm dữ liệu cho đến đường đó.



Hình 15. Minh họa hồi quy tuyến tính

Như hình minh họa, ở các điểm ở hình trên (trái) biểu diễn các điểm dữ liệu khác nhau và đường thẳng (bên phải) đại diện cho một đường gần đúng có thể giải thích mối quan hệ giữa các trục x & y. Thông qua, hồi quy tuyến tính chúng ta cố gắng tìm ra một đường như vậy. Ví dụ, nếu chúng ta có một biến phụ thuộc Y và một biến độc lập X - mối quan hệ giữa X và Y có thể được biểu diễn dưới dạng phương trình sau:

Y = Β0 + Β1\*X

Trong đó:

Y = Biến phụ thuộc

X = biến độc lập

Β0 = Hằng số

Β1 = Hệ số mối quan hệ giữa X và Y

## 2.3 Phương pháp đánh giá thuật toán

### 2.3.1 R-Squared (R²):

R-Squared là một phép đo thường được sử dụng để đánh giá mức độ phù hợp của một mô hình hồi quy tuyến tính với dữ liệu thực tế.

Giá trị R² nằm trong khoảng từ 0 đến 1. Giá trị càng gần 1 cho thấy mô hình giải thích một phần lớn sự biến động của dữ liệu. Ngược lại, giá trị càng gần 0 cho thấy mô hình không giải thích được sự biến động của dữ liệu.

Công thức tính R² thường được biểu diễn như sau:



Hình 17. Công thức tính R²

Trong đó:

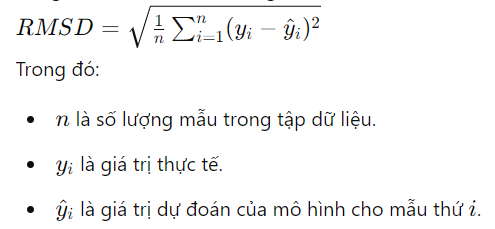
* ESS là viết tắt của Residual Sum of Squares, tức là tổng các độ lệch bình phương của phần dư.
* TSS là viết tắt của Total Sum of Squares, tức là tổng độ lệch bình phương của toàn bộ các nhân tố nghiên cứu.

### 2.3.2 Root Mean Square Deviation (RMSD):

RMSD là một phép đo để đánh giá độ chính xác của một mô hình dự đoán. Nó biểu thị sự chênh lệch trung bình giữa giá trị dự đoán và giá trị thực tế.

* Giá trị RMSD càng thấp thì mô hình càng chính xác.

Công thức tính RMSD thường được biểu diễn như sau:



Hình 18. Công thức tính RMSD

R-Squared và RMSD là hai phép đo quan trọng để đánh giá hiệu suất của một mô hình hồi quy tuyến tính trên dữ liệu thực tế. R-Squared giúp hiểu được tỷ lệ biến động dữ liệu mà mô hình có thể giải thích, trong khi RMSD cho biết mức độ chính xác trung bình của dự đoán so với giá trị thực tế.

### 2.3.3 So sánh giá trị dự đoán với giá trị thực tế:

Trong quá trình triển khai thuật toán với Pyspark, chúng em đã đối mặt với hạn chế là không được phép sử dụng các thư viện bên ngoài như sklearn.metrics để đánh giá mô hình. Để giải quyết vấn đề này, nhóm của chúng em đã xây dựng một hàm để tính độ chính xác một cách đơn giản nhưng hiệu quả.

Ví dụ một hàm **cal\_accuracy** :

A screen shot of a computer code

Description automatically generated

Hình 19. Ví dụ một hàm cal\_accuracy

Hàm **cal\_accuracy** nhận vào một DataFrame **df** chứa dữ liệu cần đánh giá, cùng với tên của cột chứa giá trị thực tế (**actual**) và tên của cột chứa giá trị dự đoán (**predict**). Sau đó, nó tính toán tỷ lệ phần trăm của các dự đoán chính xác bằng cách so sánh giữa giá trị thực tế và giá trị dự đoán.

# CHƯƠNG 3: TRIỂN KHAI MÔ HÌNH

## 3.1 Tiền xử lý dữ liệu

* Cài đặt pySpark

A close up of a white background

Description automatically generated

Hình 20. Cài đặt pySpark

* Khởi tạo spark session

A screen shot of a computer

Description automatically generated

Hình 21. Khởi tạo spark session

* Đọc dữ liệu từ file csv

A close-up of a computer code

Description automatically generated

Hình 22. Đọc dữ liệu từ file csv

* Kiểm tra dữ liệu

A screenshot of a computer program

Description automatically generated

Hình 23. Kiểm tra dữ liệu

* Kiểm tra số dòng dữ liệu(7050)

A close-up of a computer code

Description automatically generated

Hình 24. Kiểm tra số dòng dữ liệu

* Kiểm tra cấu trúc dữ liệu

A screenshot of a computer program

Description automatically generated

Hình 25. Kiểm tra cấu trúc dữ liệu

* Kiểm tra có tồn tại dữ liệu null không bằng cách đếm dữ liệu không null

A computer screen shot of a computer code

Description automatically generated

Hình 26. Kiểm tra có tồn tại dữ liệu null không bằng cách đếm dữ liệu không null

* Kiểm tra số cột dữ liệu

A screenshot of a computer screen

Description automatically generated

Hình 27. Kiểm tra số cột dữ liệu

* Xóa các cột dữ liệu thừa ‘Column1’, ‘Column2’, ‘Column3’, ‘Column4’ vì các cột này không có giá trị tính toán

A screenshot of a computer

Description automatically generated

Hình 28. Xóa các cột dữ liệu thừa

* Kiểm tra số cột sau khi xóa(12)

A close-up of a computer code

Description automatically generated

Hình 29. Kiểm tra số cột sau khi xóa

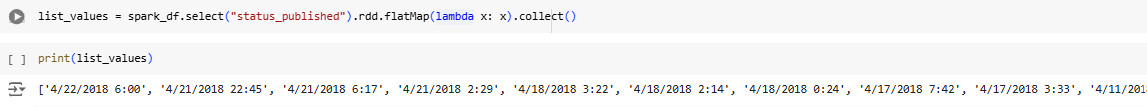
* Kiểm tra các giá trị duy nhất của cột ‘status\_type’

A screenshot of a computer program

Description automatically generated

Hình 30. Kiểm tra các giá trị duy nhất

* Lấy dữ liệu của cột “status\_published” và chuyển thành danh sách kiểu Python và in ra



Hình 31. Lấy dữ liệu của cột “status\_published” và chuyển thành danh sách kiểu Python

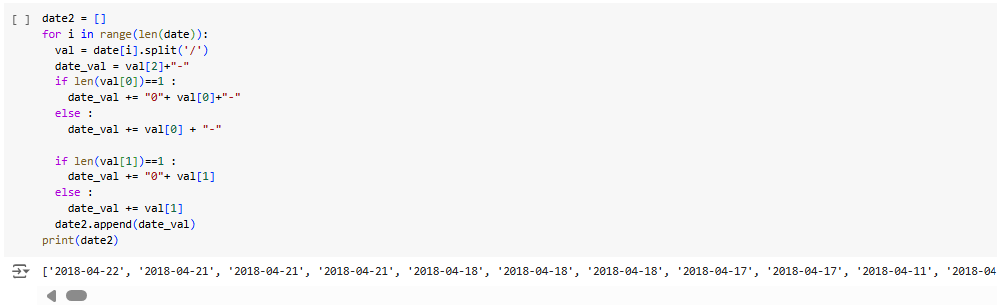
* Tách ngày và giờ bởi giá trị ‘ ’ và lưu vào hai list tên “date” và “time”

A white background with black text

Description automatically generated

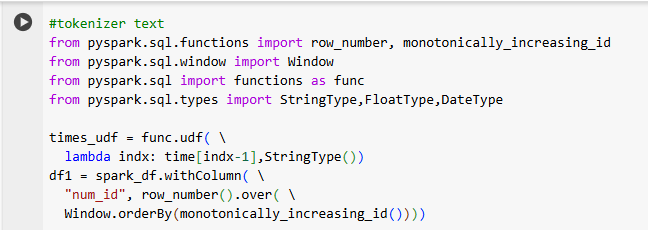
Hình 32. Tách ngày và giờ

* Định dạng lại kiểu dữ liệu ngày từ “MM/DD/YYYY” thành “YYYY-MM-DD”



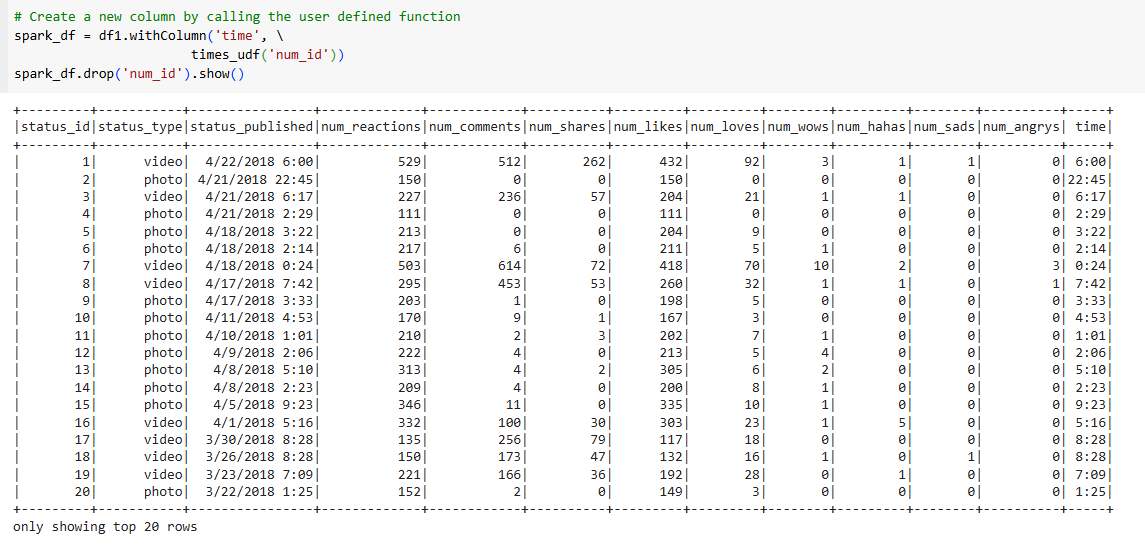
Hình 33. Định dạng lại kiểu dữ liệu ngày

* Import các thư viện cần thiết và khởi tạo hàm “times\_udf”, hàm này nhận một id và trả về giá trị thời gian tại cùng id đó trong list “time”. Bên cạnh đó khởi tạo cột “num\_id” trong dataframe “df1” để dung làm cột index tạm thời



Hình 34. Import các thư viện và khởi tạo hàm cần thiết

* Tạo một cột mới tên là “time” trong spark\_df DataFrame bằng hàm “times\_udf”. Drop cột “num\_id” sau khi đã hoàn tất và hiển thị kết quả



Hình 35. Tạo một cột mới chứa giờ

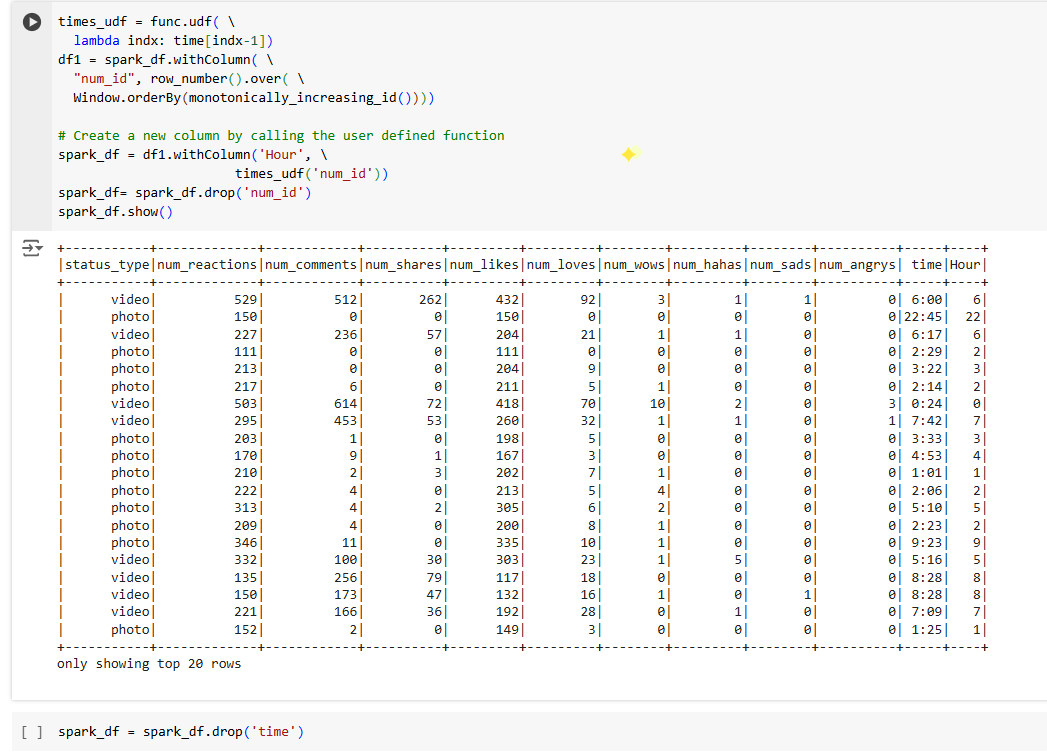
* Lấy dữ liệu giờ từ cột “time” đưa vào một list

A screenshot of a computer

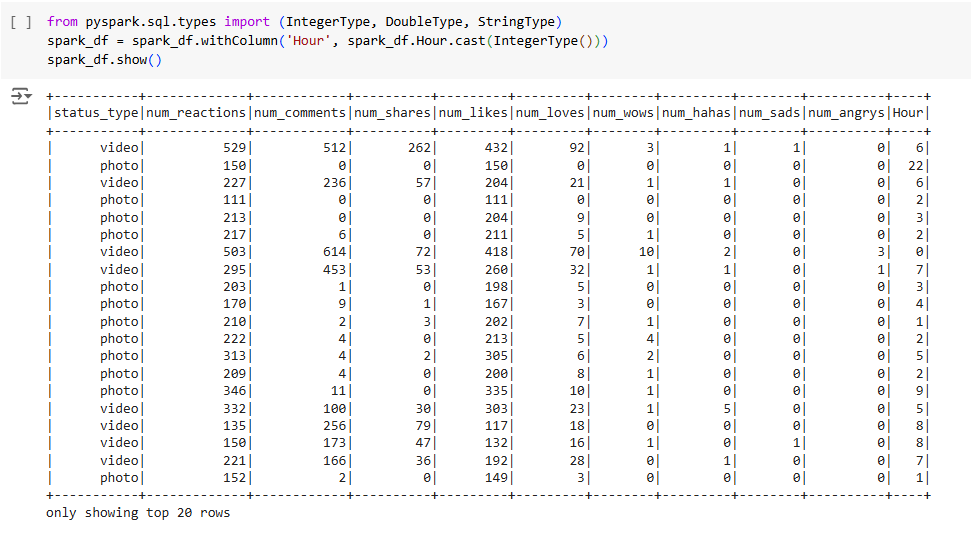
Description automatically generated

Hình 36. Lưu giờ vào một list

* Thêm các hàm cần thiết để thêm cột “Hour” chứa dữ liệu giờ vào bộ dữ liệu và hiển thị kết quả. Sau đó drop cột “time” vì không còn cần thiết cho việc dự đoán

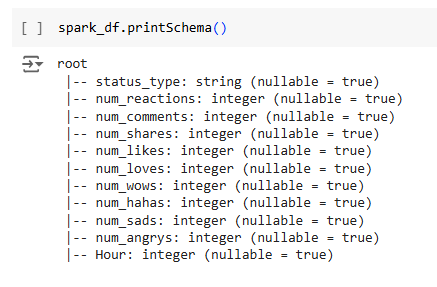
Hình 37. Tạo cột chứa dữ liệu giờ

* Chuyển kiểu dữ liệu cột “Hour” thành integer



Hình 38. Chuyển kiểu dữ liệu cột “Hour” thành integer

* Kiểm tra lại cấu trúc của bộ dữ liệu



Hình 39. Kiểm tra lại cấu trúc của bộ dữ liệu

* Import các thư viện cần thiết và thực hiện định nghĩa hàm label\_encode trên một cột dữ liệu, chuyển đổi các giá trị trên cột sang các giá trị số nguyên tuần tự

A screenshot of a computer code

Description automatically generated

Hình 40. Encoding

* Thực hiện hàm label\_encode trên cột “status\_type” và hiển thị kết quả

A screenshot of a computer program

Description automatically generated

Hình 41. Triển khai trên cột “status\_type”

* Hiển thị dữ liệu sau khi đã hoàn tất tiền xử lý

A screenshot of a computer

Description automatically generated

Hình 42. Dữ liệu sau khi tiền xử lý

Tiền xử lý dữ liệu đến đây là kết thúc, đến đây có thể bắt đầu thực hiện các giải thuật dự đoán trên bộ dữ liệu.

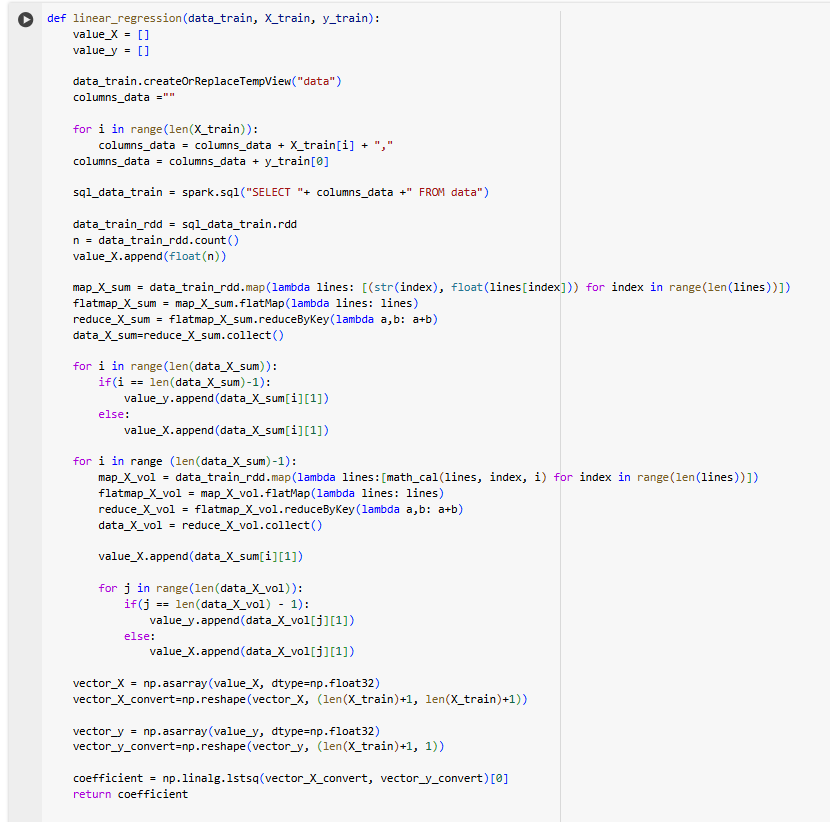
## 3.2 Triển khai mô hình

* Định nghĩa hàm math\_cal. Hàm này được sử dụng để tính toán các giá trị cần thiết cho việc thực hiện hồi quy tuyến tính



Hình 43. Định nghĩa hàm math\_cal

* Chạy thuật toán



Hình 44. Chạy thuật toán Linear Regression

Mô tả thuật toán:

1. Chuẩn bị dữ liệu:

* value\_X và value\_y là hai danh sách để lưu trữ các giá trị đầu vào (X) và đầu ra (y) của mô hình.
* Tạo một view tạm thời "data" từ dataframe data\_train.
* Tạo chuỗi columns\_data chứa tên các cột X và y.
* Sử dụng Spark SQL để truy vấn dữ liệu từ view "data" và lưu vào sql\_data\_train.
* Chuyển sql\_data\_train thành RDD (data\_train\_rdd).

1. Tính tổng các giá trị X:

* Sử dụng map() để tạo một RDD mới, với mỗi phần tử là một danh sách chứa các cặp (tên cột, giá trị) (map\_X\_sum).
* Sử dụng flatMap() để làm phẳng RDD, thành một danh sách các cặp (tên cột, giá trị) (flatmap\_X\_sum).
* Sử dụng reduceByKey() để tính tổng các giá trị cho từng cột (reduce\_X\_sum).
* Lưu trữ các tổng các giá trị X vào value\_X và tổng giá trị y vào value\_y.

1. Tính ma trận X và vector y:

* Sử dụng map() để tính các giá trị cho ma trận X, sử dụng hàm math\_cal() (không được hiển thị trong đoạn code).
* Lặp lại các bước trước để tính tổng các giá trị cho ma trận X.
* Lưu trữ các giá trị vào value\_X và value\_y.

1. Tính hệ số của mô hình hồi quy tuyến tính:

* Chuyển value\_X và value\_y thành các mảng NumPy (vector\_X và vector\_y).
* Reshape các mảng thành ma trận và vector.
* Sử dụng np.linalg.lstsq() để tìm hệ số của mô hình hồi quy tuyến tính.
* Trả về hệ số tìm được.
* Huấn luyện trên bộ dữ liệu
* Chia dữ liệu train test và sử dụng hàm linear\_regression() để xây dựng mô hình hồi quy tuyến tính

A screenshot of a computer code

Description automatically generated

Hình 45. Chia train test và ứng dụng mô hình hồi quy tuyến tính

# CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC

## 4.1 Kết quả chạy thuật toán

* Định nghĩa hàm predict

A computer code with text

Description automatically generated

Hình 46. Định nghĩa hàm predict

* Dự đoán kết quả chạy thuật toán trên tập test và hiển thị kết quả

A screenshot of a computer

Description automatically generated

Hình 47. Dự đoán trên tập test

* Chuyển đổi kết quả dự đoán về cùng kiểu dữ liệu với cột “status\_type” để tiến hành đánh giá

A screenshot of a computer

Description automatically generated

Hình 48. Đổi kiểu dữ liệu

* Tạo cột “type\_predict” chứa các giá trị kết quả

A screenshot of a computer

Description automatically generated

Hình 49. Tạo cột mới chứa kết quả dự đoán

## 4.2 Đánh giá

* Định nghĩa các độ đo

A screenshot of a computer code

Description automatically generated

Hình 50. Hàm khởi tạo R Square

A close up of a text

Description automatically generated

Hình 51. Hàm khởi tạo RMSD

A screen shot of a computer code

Description automatically generated

Hình 52. Khởi tạo hàm cal\_accuracy

* Hiển thị hai cột “status\_type” và “type\_predict” để tiến hành so sánh kết quả

A screenshot of a computer

Description automatically generated

Hình 53. Dữ liệu gốc và dữ liệu dự đoán

* Tính giá trị các độ đo

A close-up of a computer code

Description automatically generated

Hình 54. Tính R Square và RMSD

* Hệ số xác định (R-squared) có giá trị âm là -1.7610793131643003. Điều này cho thấy mô hình không phù hợp với dữ liệu, nghĩa là mô hình không giải thích được sự biến thiên của dữ liệu thực tế.
* Căn Trung Bình Bình Phương Sai số (RMSD) có giá trị là 1.1799185552809412. Với giá trị RMSD này, có thể nói rằng mô hình dự đoán chưa đạt được độ chính xác mong muốn.
* Tính giá trị accuracy

A close-up of a word

Description automatically generated

Hình 55. Tính accuracy

# CHƯƠNG 5: KẾT LUẬN

## 5.1 Ưu điểm và hạn chế

### 5.1.1 Ưu điểm

Có khả năng nhận diện, phát biểu bài toán khai thác dữ liệu lớn và áp dụng được các giải thuật khai thác dữ liệu Linear Regression vào tập dữ liệu lớn.

Dựa trên dữ liệu phân tích để có cái nhìn sâu sắc hơn về tập dữ liệu, rút ra các nhận xét về xu hướng bán hàng qua mạng xã hội, cụ thể là Facebook Live, một kênh bán hang qua mạng lớn.

Đề tài phân tích dữ liệu bán hàng có tính thực tiễn cao, có thể áp dụng trực tiếp vào chiến lược kinh doanh của nhà cung cấp nhỏ lẻ, giúp nâng cao hiệu suất, doanh thu bán hang.

### 5.1.2 Hạn chế

Nhóm còn nhiều thiếu sót về kiến thức liên quan đến khai thác phân tích dữ liệu lớn.

Dữ liệu về Facebook Live Sellers có thể có nhiều tính chất phức tạp, như mối quan hệ không tuyến tính, biến ẩn, dữ liệu bị thiếu hoặc lỗi

Linear Regression có thể không đủ mạnh để mô hình hóa các mối quan hệ phức tạp trong dữ liệu Facebook Live Sellers.

## 5.2 Hướng phát triển

Trao dồi thêm kiến thức và kinh nghiệm trong lĩnh vực dữ liệu lớn, tối ưu hóa các thuật toán và kết hợp các phương pháp, giải thuật khác để tăng độ tin cậy

Tìm kiếm bộ dữ liệu phù hợp hơn với mục tiêu của nhóm để thực hiện khai thác tốt hơn.

# PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên**  **Công việc** | **Phong** | **Hải** | **Nhân** | **Đăng** |
| Chọn đề tài, dataset | **X** | **X** | **X** | **X** |
| Mô tả bài toán | **X** | **X** | **X** | **X** |
| Mô tả dataset |  |  | **X** | **X** |
| Cơ sở lý thuyết về tiền xử lý |  | **X** |  |  |
| Tìm hiểu thuật toán áp dụng |  | **X** | **X** |  |
| Tìm hiểu phương pháp đánh giá |  | **X** |  |  |
| Tiền xử lý dữ liệu | **X** | **X** | **X** | **X** |
| Triển khai mô hình | **X** |  |  |  |
| Trực quan hóa kết quả |  |  | **X** |  |
| Đánh giá kết quả |  | **X** | **X** | **X** |
| Kết luận |  |  |  | **X** |
| Chỉnh sửa báo cáo |  | **X** |  | **X** |
| Viết và chỉnh sửa slide | **X** | **X** | **X** | **X** |
| Demo | **X** |  |  |  |
| Thuyết trình | **X** | **X** | **X** | **X** |
| **Đánh giá** | **100%** | **100%** | **100%** | **100%** |

Bảng 2. Phân công công việc

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | A. GAURAV, "Facebook Live sellers in Thailand, UCI ML Repo," 2018. [Online]. Available: https://tinyurl.com/2ndxydd5. [Accessed 2024]. |
| [2] | scikit-learn, "LinearRegression," 2024. [Online]. Available: https://tinyurl.com/2tubjzzs. [Accessed 2024]. |

Link:

<https://vi.wikipedia.org/wiki/H%E1%BB%93i_quy_tuy%E1%BA%BFn_t%C3%ADnh>

<https://viblo.asia/p/giai-thich-va-ung-dung-cua-pysparksqlwindow-trong-xu-ly-du-lieu-phan-tan-gwd43M73LX9>