

TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN
HỌC PHẦN: DỮ LIỆU LỚN
ĐỀ TÀI: DỰ ĐOÁN XU HƯỚNG ĐẶT PHÒNG KHÁCH SẠN

Giảng viên: Trần Quý Nam, Lê Thị Thùy Trang

<i>TT</i>	<i>Mã SV</i>	<i>Họ và Tên</i>	<i>Ngày Sinh</i>	<i>Lớp</i>
<i>1</i>	<i>1671020125</i>	<i>Vũ Khánh Hoàn</i>	<i>30/08/2004</i>	<i>CNTT 16-02</i>
<i>2</i>	<i>1671020117</i>	<i>Vũ Văn Hiệp</i>	<i>25/01/2004</i>	<i>CNTT 16-02</i>
<i>4</i>	<i>1671020190</i>	<i>Nguyễn Ngọc Bảo Long</i>	<i>26/03/2004</i>	<i>CNTT 16-02</i>
<i>5</i>	<i>1671020049</i>	<i>Nguyễn Ánh Cường</i>	<i>16/06/2004</i>	<i>CNTT 16-02</i>

Hà Nội, năm 2025

TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN
HỌC PHẦN: DỮ LIỆU LỚN
ĐỀ TÀI: DỰ ĐOÁN XU HƯỚNG ĐẶT PHÒNG KHÁCH SẠN

<i>TT</i>	<i>Mã SV</i>	<i>Họ và Tên</i>	<i>Ngày Sinh</i>	<i>Điểm</i>	
				<i>Bảng Số</i>	<i>Bảng Chữ</i>
<i>1</i>	<i>1671020125</i>	<i>Vũ Khánh Hoàn</i>	<i>30/08/2004</i>		
<i>2</i>	<i>1671020117</i>	<i>Vũ Văn Hiệp</i>	<i>25/01/2004</i>		
<i>4</i>	<i>1671020190</i>	<i>Nguyễn Ngọc Bảo Long</i>	<i>26/03/2004</i>		
<i>5</i>	<i>1671020049</i>	<i>Nguyễn Ánh Cường</i>	<i>16/06/2004</i>		

Hà Nội, năm 2025

LỜI NÓI ĐẦU

Trong bối cảnh công nghệ số đang phát triển mạnh mẽ, dữ liệu lớn (Big Data) đã và đang trở thành một trong những yếu tố cốt lõi giúp các doanh nghiệp tối ưu hóa quy trình hoạt động, nâng cao chất lượng dịch vụ và đưa ra các quyết định kinh doanh chính xác. Trong ngành du lịch và khách sạn, việc khai thác dữ liệu đặt phòng không chỉ giúp doanh nghiệp nắm bắt xu hướng thị trường mà còn hỗ trợ dự báo nhu cầu, tối ưu hóa giá phòng, quản lý nguồn lực và gia tăng trải nghiệm khách hàng.

Xuất phát từ thực tế đó, chúng em quyết định chọn đề tài "Dự đoán xu hướng đặt phòng khách sạn" nhằm ứng dụng công nghệ Big Data và Sparklyr để phân tích, xây dựng mô hình dự đoán khả năng đặt phòng của khách hàng. Sparklyr là một công cụ mạnh mẽ giúp tích hợp Apache Spark với R, cho phép xử lý dữ liệu lớn một cách nhanh chóng và hiệu quả. Thông qua việc khai thác tập dữ liệu đặt phòng thực tế, đề tài này sẽ tập trung vào việc tìm hiểu các yếu tố ảnh hưởng đến quyết định đặt phòng của khách hàng và từ đó đề xuất mô hình dự đoán phù hợp.

Báo cáo được xây dựng dựa trên quá trình nghiên cứu, phân tích dữ liệu, lựa chọn mô hình dự đoán và đánh giá kết quả. Nội dung chính của báo cáo bao gồm ba chương:

- Chương 1: Tổng quan về công nghệ và phương pháp, trình bày các khái niệm cơ bản về Big Data, Sparklyr và các phương pháp dự đoán xu hướng đặt phòng khách sạn.
- Chương 2: Xây dựng mô hình dự đoán, bao gồm các bước tiền xử lý dữ liệu, phân tích, lựa chọn đặc trưng và huấn luyện mô hình.
- Chương 3: Kết quả và đánh giá, phân tích hiệu suất mô hình, so sánh kết quả với các phương pháp khác và đề xuất hướng phát triển trong tương lai.

Trong suốt quá trình thực hiện đề tài, chúng em đã gặp không ít khó khăn, từ việc tìm hiểu về Sparklyr, xử lý dữ liệu lớn cho đến việc lựa chọn mô hình dự đoán phù hợp. Tuy nhiên, nhờ sự hướng dẫn tận tình của thầy/cô cùng sự hỗ trợ từ các tài liệu nghiên cứu, chúng em đã có thể hoàn thành bài tập lớn này.

Chúng em xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn, những người đã dành thời gian hỗ trợ và đóng góp ý kiến quý báu để đề tài được hoàn thiện. Đồng thời,

chúng em cũng trân trọng cảm ơn các nguồn tài liệu tham khảo đã giúp chúng em có thêm kiến thức và định hướng trong quá trình thực hiện nghiên cứu này.

Mặc dù đã cố gắng hoàn thiện bài báo cáo một cách tốt nhất, nhưng do thời gian và kinh nghiệm còn hạn chế, chắc chắn vẫn còn những thiếu sót. Chúng em mong nhận được những góp ý từ thầy/cô để có thể cải thiện và nâng cao hơn nữa chất lượng nghiên cứu.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

LỜI NÓI ĐẦU	0
MỤC LỤC	2
DANH MỤC HÌNH ẢNH	4
DANH MỤC BẢNG BIỂU	5
CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ VÀ PHƯƠNG PHÁP	6
1.1. Giới thiệu về Big Data và Sparklyr	6
1.1.1. Khái niệm về Big Data và ứng dụng trong dự đoán xu hướng	6
1.1.2. Giới thiệu về Sparklyr và lý do sử dụng trong bài toán này	12
1.2. Các phương pháp dự đoán xu hướng đặt phòng khách sạn	16
1.2.1. Các yếu tố ảnh hưởng đến quyết định đặt phòng	16
1.2.2. Các phương pháp phổ biến trong dự đoán xu hướng đặt phòng	17
1.3. Tổng quan về tập dữ liệu	19
1.3. Giới thiệu về nguồn dữ liệu	19
1.3.1. Mô tả các thuộc tính trong dữ liệu	19
1.3.2. Xử lý dữ liệu bị thiếu và dữ liệu không hợp lệ	19
CHƯƠNG 2. XÂY DỰNG MÔ HÌNH DỰ ĐOÁN	20
2.1. Giới thiệu tập dữ liệu đặt phòng khách sạn	20
2.2. Nhập thư viện	21
2.3. Tải dữ liệu	22
2.4. Khám phá dữ liệu	22
2.5. Tiền xử lý dữ liệu	25
2.6. Phân tích và trực quan hoá dữ liệu	26
2.7. Mô hình dự đoán	31

2.7.1. Chuẩn bị mô hình	31
2.7.2. Mô hình cây quyết định	32
2.7.3. Mô hình rừng ngẫu nhiên	34
CHƯƠNG 3. KẾT QUẢ VÀ ĐÁNH GIÁ	36
3.1. Hiệu suất mô hình	36
3.2. Phân tích và so sánh kết quả	37
3.2.1. Các yếu tố ảnh hưởng đến kết quả dự đoán	37
3.2.2. So sánh với các phương pháp truyền thống	39
3.2.3. Tổng kết và đề xuất	40
KẾT LUẬN	42

DANH MỤC HÌNH ẢNH

Hình 1. Big Data	6
Hình 2. 7V – bảy đặc điểm chính của Big Data	7
Hình 3. Lịch sử phát triển của Apache Spark	12
Hình 4. Thành phần của Spark	13
Hình 5. Những doanh nghiệp sử dụng Apache Spark	15
Hình 6. Nhập thư viện	21
Hình 7. Khởi tạo một phiên làm việc với Spark	22
Hình 8. Hiển thị 5 dòng đầu tiên để kiểm tra dữ liệu	22
Hình 9. Kiểm tra cấu trúc dữ liệu	23
Hình 10. Đếm số lượng dòng và cột	23
Hình 11. Các thống kê cơ bản	24
Hình 12. Kiểm tra số lượng giá trị null trong mỗi cột	24
Hình 13. Biểu đồ hiển thị giá trị thiếu	24
Hình 14. Kiểm tra kiểu dữ liệu của các cột trong DataFrame	25
Hình 15. Thay thế các giá trị thiếu trong cột children bằng 0	26
Hình 16. Trạng thái đặt phòng	26
Hình 17. Biểu đồ Top 10 quốc gia có số lượng đặt phòng bị hủy nhiều nhất	27
Hình 18. Biểu đồ tỷ lệ đặt phòng theo từng tháng trong năm	29
Hình 19. Biểu đồ tỷ lệ hủy đặt phòng theo từng phân khúc thị trường	30
Hình 20. Mô hình cây quyết định	32
Hình 21. Kết quả mô hình cây quyết định	33
Hình 22. Mô hình rừng ngẫu nhiên	34
Hình 23. Kết quả phân tích theo mô hình rừng ngẫu nhiên	34

DANH MỤC BẢNG BIỂU

Bảng 1. 7V – bảy đặc điểm chính của Big Data	7
Bảng 2. Hệ thống lưu trữ và xử lý dữ liệu	9
Bảng 3. Hệ thống quản lý cơ sở dữ liệu NoSQL	9
Bảng 4. Công nghệ thu thập và xử lý dữ liệu streaming	10
Bảng 5. Công cụ phân tích và trực quan hóa dữ liệu	10
Bảng 6. Trí tuệ nhân tạo (AI) và Machine Learning (ML) cho Big Data	11
Bảng 7. Bảng So sánh hiệu suất giữa phương pháp học máy và truyền thống	39

CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ VÀ PHƯƠNG PHÁP

1.1. Giới thiệu về Big Data và Sparklyr

1.1.1. Khái niệm về Big Data và ứng dụng trong dự đoán xu hướng

a. Khái niệm

Dữ liệu lớn (Big Data) là thuật ngữ dùng để chỉ khối lượng dữ liệu khổng lồ, phức tạp đến mức các công cụ quản lý dữ liệu truyền thống không thể thu thập, xử lý và phân tích một cách hiệu quả trong thời gian hợp lý.

Big Data không chỉ đơn thuần đề cập đến kích thước dữ liệu mà còn bao gồm các yếu tố quan trọng như tốc độ xử lý, sự đa dạng và tính xác thực của dữ liệu. Các tập dữ liệu lớn này có thể bao gồm dữ liệu có cấu trúc, không có cấu trúc và bán cấu trúc, được khai thác để tìm ra những thông tin có giá trị (insights). Mặc dù không có tiêu chuẩn cố định về kích thước của Big Data, nhưng thường được tính theo đơn vị petabyte hoặc thậm chí exabyte trong các dự án quy mô lớn.

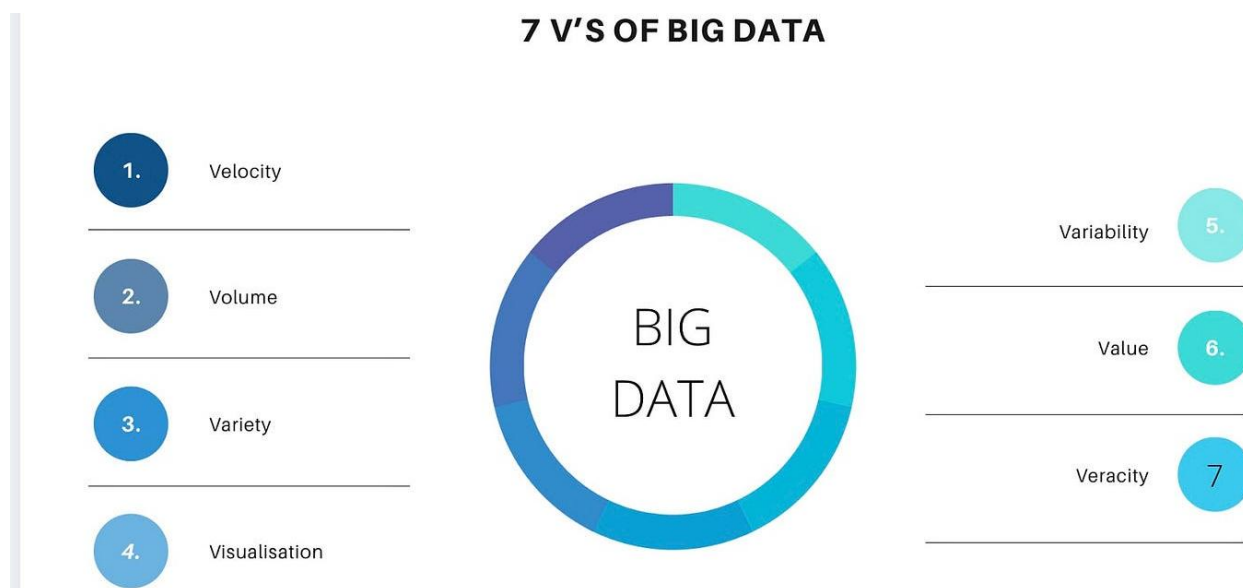


Hình 1. Big Data

Nguồn dữ liệu cho Big Data rất đa dạng, có thể đến từ các trang web, mạng xã hội, ứng dụng di động và máy tính, các thí nghiệm khoa học, cũng như từ các thiết bị cảm biến và hệ thống Internet of Things (IoT).

b. Các đặc điểm và tính chất quan trọng của Big Data

Big Data không chỉ đơn thuần là dữ liệu có kích thước lớn mà còn có nhiều đặc điểm quan trọng giúp chúng ta hiểu rõ hơn về cách quản lý, phân tích và khai thác dữ liệu hiệu quả. Dưới đây là 7V – bảy đặc điểm chính của Big Data:



Hình 2. 7V – bảy đặc điểm chính của Big Data

Để mô tả chi tiết và ngắn gọn nhất 7 tính chất của Big Data, chúng em đã mô phỏng dưới dạng bảng sau:

Bảng 1. 7V – bảy đặc điểm chính của Big Data

STT	Đặc điểm	Mô tả	Ví dụ thực tế
1	Volume (Khối lượng)	Dữ liệu trong Big Data có khối lượng rất lớn, có thể lên đến hàng terabyte (TB), petabyte (PB) hoặc exabyte (EB). Cần hệ thống lưu trữ và xử lý mạnh mẽ.	Dữ liệu từ Facebook, Google, Amazon chứa hàng petabyte thông tin mỗi ngày.
2	Velocity (Tốc độ)	Dữ liệu được tạo ra và truyền tải với tốc độ nhanh, đòi hỏi phải xử lý theo thời gian thực hoặc gần thời gian thực.	Cập nhật thị trường chứng khoán, dữ liệu IoT từ cảm

			biến, luồng tin tức mạng xã hội.
3	Variety (Đa dạng)	Dữ liệu có nhiều loại định dạng khác nhau: có cấu trúc, bán cấu trúc, phi cấu trúc (văn bản, hình ảnh, video, âm thanh, log files, dữ liệu IoT).	Email, tin nhắn, hình ảnh từ Instagram, dữ liệu từ cảm biến xe hơi.
4	Veracity (Tính xác thực)	Dữ liệu có thể chứa lỗi, không nhất quán, bị nhiễu, cần làm sạch và xác thực để đảm bảo độ tin cậy.	Dữ liệu sai lệch từ mạng xã hội, tin giả (fake news), dữ liệu cảm biến bị nhiễu.
5	Value (Giá trị)	Giá trị của dữ liệu phụ thuộc vào khả năng phân tích và khai thác để đưa ra quyết định chiến lược.	Phân tích hành vi khách hàng, dự đoán xu hướng thị trường, tối ưu hóa chuỗi cung ứng.
6	Variability (Biến động)	Dữ liệu không ổn định, có thể thay đổi theo thời gian hoặc theo ngữ cảnh, cần hệ thống linh hoạt và thích ứng.	Xu hướng tìm kiếm trên Google thay đổi theo sự kiện, phản ứng của khách hàng theo mùa.
7	Visualization (Trực quan hóa)	Dữ liệu phức tạp cần được trực quan hóa bằng đồ thị, biểu đồ để dễ hiểu và hỗ trợ ra quyết định.	Báo cáo bằng Power BI, Tableau, Google Data Studio, dashboard kinh doanh.

c. Các công nghệ đặc biệt dành cho Big Data

Big Data yêu cầu các công nghệ mạnh mẽ để thu thập, lưu trữ, xử lý và phân tích dữ liệu khổng lồ. Dưới đây là những công nghệ quan trọng trong hệ sinh thái Big Data:

1. Hệ thống lưu trữ và xử lý dữ liệu

Bảng 2. Hệ thống lưu trữ và xử lý dữ liệu

Công nghệ	Mô tả	Ứng dụng thực tế
Hadoop	Framework mã nguồn mở để lưu trữ và xử lý dữ liệu lớn theo mô hình MapReduce.	Lưu trữ và phân tích dữ liệu phi cấu trúc trong các doanh nghiệp lớn.
Apache Spark	Công cụ xử lý dữ liệu nhanh, hỗ trợ xử lý thời gian thực và tính toán phân tán tốt hơn Hadoop.	Phân tích dữ liệu tài chính, dự báo xu hướng khách hàng.
Apache Flink	Nền tảng xử lý dữ liệu streaming (luồng dữ liệu) theo thời gian thực.	Giám sát mạng, phát hiện gian lận.
Apache Storm	Hệ thống xử lý luồng dữ liệu phân tán theo thời gian thực.	Phân tích dữ liệu mạng xã hội, IoT.

2. Hệ thống quản lý cơ sở dữ liệu NoSQL

Bảng 3. Hệ thống quản lý cơ sở dữ liệu NoSQL

Công nghệ	Mô tả	Ứng dụng thực tế
MongoDB	Cơ sở dữ liệu NoSQL dạng tài liệu, phù hợp với dữ liệu phi cấu trúc.	Lưu trữ dữ liệu người dùng từ các ứng dụng web, IoT.
Cassandra	Cơ sở dữ liệu phân tán, tốc độ cao, hỗ	Xử lý dữ liệu thời gian thực từ

	trợ scaling ngang tốt.	các hệ thống giao dịch.
HBase	Cơ sở dữ liệu NoSQL chạy trên Hadoop, tối ưu cho xử lý dữ liệu lớn.	Hệ thống phân tích dữ liệu lớn trong viễn thông.
Elasticsearch	Công cụ tìm kiếm và phân tích dữ liệu dạng full-text search.	Công cụ tìm kiếm trong các website thương mại điện tử.

3. Công nghệ thu thập và xử lý dữ liệu streaming

Bảng 4. Công nghệ thu thập và xử lý dữ liệu streaming

Công nghệ	Mô tả	Ứng dụng thực tế
Apache Kafka	Nền tảng stream processing phổ biến, xử lý dữ liệu theo thời gian thực.	Hệ thống xử lý log, giám sát hoạt động người dùng.
RabbitMQ	Hệ thống hàng đợi tin nhắn (Message Queue), hỗ trợ giao tiếp giữa các dịch vụ.	Hệ thống thông báo, chat real-time.
Google Cloud Pub/Sub	Dịch vụ pub/sub trên nền tảng Google Cloud, hỗ trợ truyền tải dữ liệu lớn.	Xử lý dữ liệu IoT, AI real-time.

4. Công cụ phân tích và trực quan hóa dữ liệu

Bảng 5. Công cụ phân tích và trực quan hóa dữ liệu

Công nghệ	Mô tả	Ứng dụng thực tế
Tableau	Công cụ trực quan hóa dữ liệu mạnh mẽ.	Tạo dashboard cho doanh nghiệp.
Power BI	Công cụ phân tích và báo cáo dữ liệu	Báo cáo kinh doanh, phân tích

	của Microsoft.	dữ liệu khách hàng.
Google Data Studio	Công cụ trực quan hóa dữ liệu miễn phí của Google.	Phân tích dữ liệu marketing, SEO.
Apache Superset	Nền tảng BI mã nguồn mở, hỗ trợ phân tích dữ liệu lớn.	Báo cáo dữ liệu doanh nghiệp với Big Data.

5. Trí tuệ nhân tạo (AI) và Machine Learning (ML) cho Big Data

Bảng 6. Trí tuệ nhân tạo (AI) và Machine Learning (ML) cho Big Data

Công nghệ	Mô tả	Ứng dụng thực tế
TensorFlow	Thư viện mã nguồn mở cho học máy và deep learning.	Nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên.
PyTorch	Framework học sâu phổ biến, dễ sử dụng.	Phát hiện gian lận, AI chatbot.
MLlib (Apache Spark)	Thư viện machine learning chạy trên Apache Spark.	Dự đoán khách hàng rời bỏ, phân tích hành vi người dùng.
H2O.ai	Nền tảng AI và ML cho doanh nghiệp.	Phân tích tài chính, dự đoán rủi ro.

Big Data không thể hoạt động hiệu quả nếu không có sự hỗ trợ của các công nghệ tiên tiến. Việc kết hợp các công nghệ trên sẽ giúp doanh nghiệp khai thác dữ liệu lớn một cách tối ưu, từ lưu trữ, xử lý đến phân tích và trực quan hóa dữ liệu.

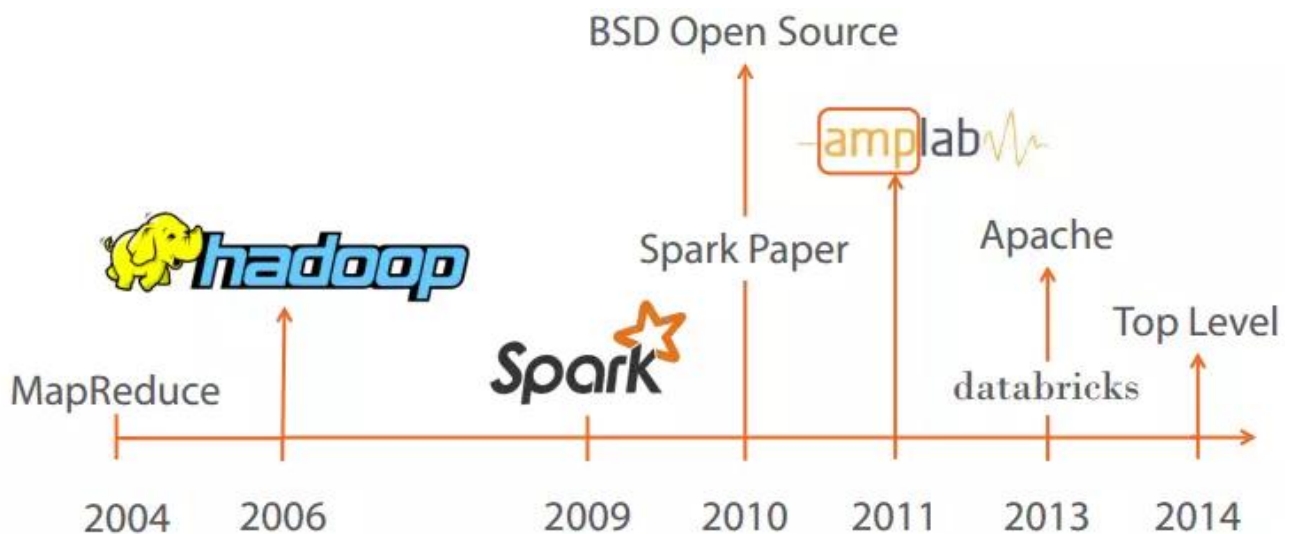
d. Ứng dụng của Big Data trong dự đoán xu hướng

Big Data đã và đang được ứng dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là trong ngành du lịch và khách sạn. Một số ứng dụng quan trọng bao gồm:

- Dự đoán nhu cầu đặt phòng: Dựa vào dữ liệu lịch sử, có thể xây dựng mô hình dự đoán số lượng đặt phòng trong tương lai, giúp khách sạn tối ưu hóa nguồn lực.
- Cá nhân hóa dịch vụ khách hàng: Phân tích dữ liệu đặt phòng và hành vi khách hàng để đưa ra các gợi ý phù hợp, nâng cao trải nghiệm người dùng.
- Phân tích xu hướng thị trường: Sử dụng dữ liệu lớn để xác định các yếu tố ảnh hưởng đến quyết định đặt phòng như giá cả, mùa du lịch, đánh giá từ khách hàng, v.v.
- Tối ưu hóa chiến lược kinh doanh: Giúp khách sạn đưa ra các chính sách giá linh hoạt dựa trên nhu cầu thực tế.

1.1.2. Giới thiệu về Sparklyr và lý do sử dụng trong bài toán này

a. Giới thiệu về Apache Spark



Hình 3. Lịch sử phát triển của Apache Spark

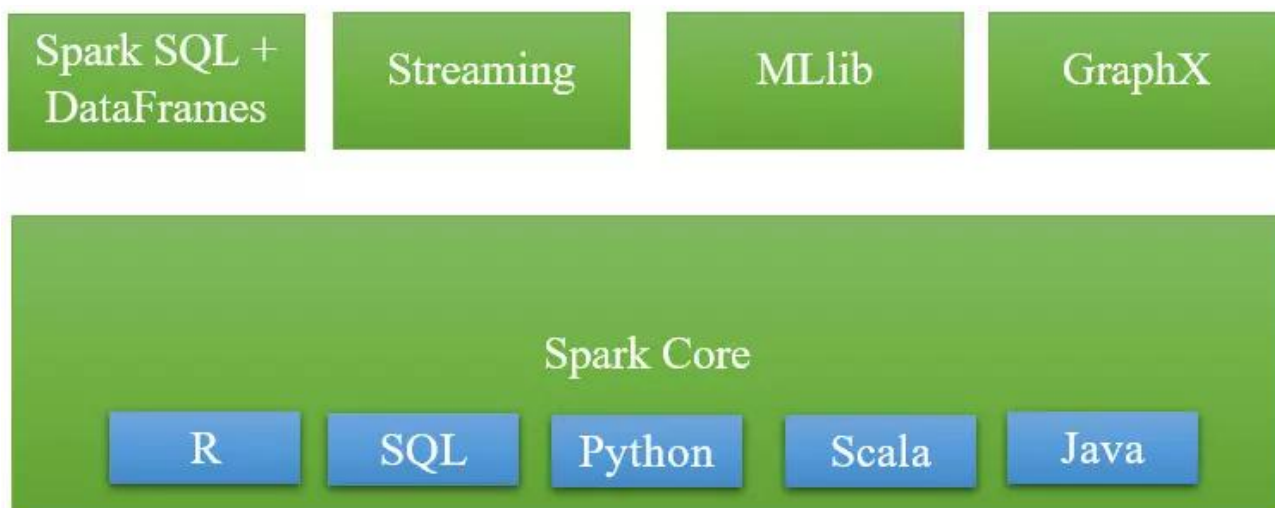
Apache Spark là một framework mã nguồn mở tính toán cụm, được phát triển sơ khởi vào năm 2009 bởi AMPLab. Sau này, Spark đã được trao cho Apache Software Foundation vào năm 2013 và được phát triển cho đến nay.

Tốc độ xử lý của Spark có được do việc tính toán được thực hiện cùng lúc trên nhiều máy khác nhau. Đồng thời việc tính toán được thực hiện ở bộ nhớ trong (in-memories) hay thực hiện hoàn toàn trên RAM.

Spark cho phép xử lý dữ liệu theo thời gian thực, vừa nhận dữ liệu từ các nguồn khác nhau đồng thời thực hiện ngay việc xử lý trên dữ liệu vừa nhận được (Spark Streaming).

Spark không có hệ thống file của riêng mình, nó sử dụng hệ thống file khác như: HDFS, Cassandra, S3,... Spark hỗ trợ nhiều kiểu định dạng file khác nhau (text, csv, json...) đồng thời nó hoàn toàn không phụ thuộc vào bất cứ một hệ thống file nào.

b. Thành phần của Spark



Hình 4. Thành phần của Spark

Apache Spark gồm có 5 thành phần chính : Spark Core, Spark Streaming, Spark SQL, MLlib và GraphX, trong đó:

- Spark Core là nền tảng cho các thành phần còn lại và các thành phần này muốn khởi chạy được thì đều phải thông qua Spark Core do Spark Core đảm nhận vai trò thực hiện công việc tính toán và xử lý trong bộ nhớ (In-memory computing) đồng thời nó cũng tham chiếu các dữ liệu được lưu trữ tại các hệ thống lưu trữ bên ngoài.
- Spark SQL cung cấp một kiểu data abstraction mới (SchemaRDD) nhằm hỗ trợ cho cả kiểu dữ liệu có cấu trúc (structured data) và dữ liệu nửa cấu trúc (semi-structured data – thường là dữ liệu dữ liệu có cấu trúc nhưng không đồng nhất và cấu trúc của dữ liệu phụ thuộc vào chính nội dung của dữ liệu ấy). Spark SQL hỗ trợ DSL (Domain-specific language) để thực hiện các thao tác trên DataFrames bằng ngôn ngữ Scala, Java hoặc Python và nó cũng hỗ trợ cả ngôn ngữ SQL với giao diện command-line và ODBC/JDBC server.
- Spark Streaming được sử dụng để thực hiện việc phân tích stream bằng việc coi stream là các mini-batches và thực hiện kỹ thuật RDD transformation đối với các dữ

liệu mini-batches này. Qua đó cho phép các đoạn code được viết cho xử lý batch có thể được tận dụng lại vào trong việc xử lý stream, làm cho việc phát triển lambda architecture được dễ dàng hơn. Tuy nhiên điều này lại tạo ra độ trễ trong xử lý dữ liệu (độ trễ chính bằng mini-batch duration) và do đó nhiều chuyên gia cho rằng Spark Streaming không thực sự là công cụ xử lý streaming giống như Storm hoặc Flink.

- MLlib (Machine Learning Library): MLlib là một nền tảng học máy phân tán bên trên Spark do kiến trúc phân tán dựa trên bộ nhớ. Theo các so sánh benchmark Spark MLlib nhanh hơn 9 lần so với phiên bản chạy trên Hadoop (Apache Mahout).
- GrapX: Grapx là nền tảng xử lý đồ thị dựa trên Spark. Nó cung cấp các Api để diễn tả các tính toán trong đồ thị bằng cách sử dụng Pregel Api.

c. Những điểm nổi bật của Spark

- Xử lý dữ liệu: Spark xử lý dữ liệu theo lô và thời gian thực
- Tính tương thích: Có thể tích hợp với tất cả các nguồn dữ liệu và định dạng tệp được hỗ trợ bởi cụm Hadoop.
- Hỗ trợ ngôn ngữ: hỗ trợ Java, Scala, Python và R.
- Phân tích thời gian thực:
 - + Apache Spark có thể xử lý dữ liệu thời gian thực tức là dữ liệu đến từ các luồng sự kiện thời gian thực với tốc độ hàng triệu sự kiện mỗi giây. Ví dụ: Data Twitter chẳng hạn hoặc lượt chia sẻ, đăng bài trên Facebook. Sức mạnh Spark là khả năng xử lý luồng trực tiếp hiệu quả.
 - + Apache Spark có thể được sử dụng để xử lý phát hiện gian lận trong khi thực hiện các giao dịch ngân hàng. Đó là bởi vì, tất cả các khoản thanh toán trực tuyến được thực hiện trong thời gian thực và chúng ta cần ngừng giao dịch gian lận trong khi quá trình thanh toán đang diễn ra.
- Mục tiêu sử dụng:
 - + Xử lý dữ liệu nhanh và tương tác
 - + Xử lý đồ thị
 - + Công việc lặp đi lặp lại
 - + Xử lý thời gian thực

- + joining Dataset
- + Machine Learning

Apache Spark là Framework thực thi dữ liệu dựa trên Hadoop HDFS. Apache Spark không thay thế cho Hadoop nhưng nó là một framework ứng dụng. Apache Spark tuy ra đời sau nhưng được nhiều người biết đến hơn Apache Hadoop vì khả năng xử lý hàng loạt và thời gian thực.

d. Những doanh nghiệp sử dụng Apache Spark

Hiện nay, có rất nhiều hãng lớn đã dùng Spark cho các sản phẩm của mình như Yahoo, ebay, IBM, Cisco...



Hình 5. Những doanh nghiệp sử dụng Apache Spark

e. Lý do sử dụng Sparklyr trong bài toán này

Bài toán dự đoán xu hướng đặt phòng khách sạn yêu cầu xử lý một lượng dữ liệu lớn với nhiều đặc trưng khác nhau. Sparklyr được lựa chọn vì các lý do sau:

- Khả năng xử lý dữ liệu lớn: Sparklyr cho phép làm việc với các tập dữ liệu có kích thước lớn mà R thông thường không thể xử lý hiệu quả.
- Tích hợp tốt với R: Dễ dàng kết nối với các thư viện mạnh mẽ của R như ggplot2, dplyr để phân tích và trực quan hóa dữ liệu.

- Hỗ trợ mô hình học máy: Sparklyr tích hợp với Spark MLlib, giúp huấn luyện và triển khai các mô hình dự đoán trên hệ thống phân tán.
- Tăng tốc hiệu suất: Sparklyr tận dụng sức mạnh của Apache Spark để thực thi các tác vụ nhanh hơn so với các phương pháp truyền thống.
- Với những ưu điểm trên, Sparklyr là lựa chọn phù hợp để triển khai bài toán dự đoán xu hướng đặt phòng khách sạn, giúp tối ưu hiệu suất xử lý và khai thác giá trị từ dữ liệu lớn.

1.2. Các phương pháp dự đoán xu hướng đặt phòng khách sạn

Dự đoán xu hướng đặt phòng khách sạn là một bài toán quan trọng trong ngành du lịch và khách sạn. Việc nắm bắt các yếu tố ảnh hưởng và áp dụng các phương pháp phân tích dữ liệu phù hợp sẽ giúp các khách sạn tối ưu hóa chiến lược kinh doanh, nâng cao tỷ lệ đặt phòng và cải thiện trải nghiệm khách hàng.

1.2.1. Các yếu tố ảnh hưởng đến quyết định đặt phòng

a. Yếu tố kinh tế và thị trường

- Giá phòng (Average Daily Rate - ADR): Giá phòng trung bình là một trong những yếu tố quan trọng nhất ảnh hưởng đến quyết định đặt phòng của khách hàng. Khi giá tăng cao, khách hàng có xu hướng tìm kiếm các lựa chọn rẻ hơn hoặc trì hoãn đặt phòng.
- Sự biến động của thị trường du lịch: Các yếu tố như khủng hoảng kinh tế, dịch bệnh (COVID-19), hoặc sự kiện đặc biệt (World Cup, lễ hội) có thể làm thay đổi xu hướng đặt phòng.
- Chính sách giảm giá, ưu đãi: Các chương trình giảm giá, mã khuyến mãi hoặc chính sách hoàn tiền có thể tác động đến quyết định đặt phòng.

b. Yếu tố hành vi khách hàng

- Thời gian đặt phòng: Khách hàng có thể đặt phòng sớm trước nhiều tháng hoặc đặt gấp sát ngày đến. Việc phân tích khoảng thời gian đặt phòng giúp dự đoán xu hướng đặt phòng theo mùa.
- Hành vi hủy phòng: Một số khách hàng có thói quen đặt phòng nhưng hủy vào phút

chót. Việc phân tích dữ liệu hủy phòng giúp khách sạn điều chỉnh chính sách đặt cọc hoặc hoàn tiền.

- Loại khách hàng: Dữ liệu có thể chia khách hàng thành nhiều nhóm như khách du lịch, khách công tác, khách đặt theo nhóm hoặc cá nhân. Mỗi nhóm khách hàng có hành vi đặt phòng khác nhau.

c. Yếu tố liên quan đến đặc điểm khách sạn

- Loại khách sạn: Các khách sạn nghỉ dưỡng (Resort Hotel) thường có lượng đặt phòng cao hơn vào mùa du lịch, trong khi khách sạn thành phố (City Hotel) có lượng đặt phòng ổn định quanh năm.
- Đánh giá khách sạn: Các khách sạn có đánh giá cao trên các nền tảng như Booking.com, Agoda, Tripadvisor thường có tỷ lệ đặt phòng cao hơn.
- Dịch vụ đi kèm: Các tiện ích như bữa sáng miễn phí, hồ bơi, trung tâm thể dục, Wi-Fi miễn phí cũng có thể tác động đến quyết định đặt phòng.

1.2.2. Các phương pháp phổ biến trong dự đoán xu hướng đặt phòng

Việc dự đoán xu hướng đặt phòng khách sạn có thể được thực hiện thông qua nhiều phương pháp khác nhau, bao gồm phân tích dữ liệu, học máy và hồi quy. Trong phạm vi bài toán này, chúng em tập trung vào các phương pháp có thể triển khai với Sparklyr.

a. Phân tích đặc trưng (Feature Analysis) và tiền xử lý dữ liệu

- Xử lý giá trị thiếu (Missing Values): Trước khi xây dựng mô hình dự đoán, cần xác định và xử lý các giá trị thiếu trong dữ liệu để đảm bảo độ chính xác.
- Phân tích tương quan giữa các biến (Variable Similarity): Giúp xác định các mối quan hệ quan trọng giữa các biến đầu vào và kết quả dự đoán.
- Chuẩn hóa dữ liệu: Một số mô hình học máy yêu cầu dữ liệu đầu vào phải được chuẩn hóa để đảm bảo tính nhất quán.

b. Trực quan hóa dữ liệu (Visualization)

- Biểu đồ phân bố giá trị ADR: Giúp xác định khoảng giá phổ biến của phòng khách sạn và ảnh hưởng của giá đến tỷ lệ đặt phòng.

- Biểu đồ xu hướng theo thời gian: Xác định các mùa cao điểm, thấp điểm của đặt phòng khách sạn.
- Heatmap tương quan giữa các biến: Giúp nhận diện các biến quan trọng trong việc dự đoán xu hướng đặt phòng.

c. Mô hình phân loại (Classification Models)

Mục tiêu: Xác định xem một đặt phòng có khả năng bị hủy hay không, hoặc phân loại khách hàng theo nhóm hành vi đặt phòng.

Các bước chính:

- Setup: Xác định biến đầu vào, phân chia tập huấn luyện và tập kiểm tra.
- Interpretation Classification: Đánh giá tầm quan trọng của từng biến đối với kết quả phân loại.
- Prediction Classification: Huấn luyện mô hình và dự đoán kết quả.
- Các thuật toán phân loại phổ biến:
 - Random Forest: Một trong những mô hình mạnh mẽ để dự đoán khả năng đặt phòng.
 - Decision Tree: Cung cấp cách tiếp cận trực quan và dễ hiểu để xác định các yếu tố quyết định đặt phòng.
 - Support Vector Machine (SVM): Thích hợp khi dữ liệu có nhiều biến số và độ phân tán cao.

d. Mô hình hồi quy (Regression Models)

Mục tiêu: Dự đoán giá trị ADR hoặc lượng đặt phòng dự kiến dựa trên các biến đầu vào.

Các bước chính:

- Visualization of adr Variable: Phân tích sự thay đổi của ADR theo các yếu tố như mùa du lịch, loại khách sạn.
- Interpretation Regression: Đánh giá ảnh hưởng của các yếu tố như thời gian đặt phòng, số lượng khách lên ADR.

- Advanced Tools for Regression: Sử dụng các kỹ thuật nâng cao như hồi quy Ridge, Lasso để cải thiện mô hình.

Các thuật toán hồi quy phổ biến:

- Linear Regression: Dự đoán giá phòng dựa trên các biến đầu vào.
- Gradient Boosting Regression: Một thuật toán mạnh mẽ giúp cải thiện độ chính xác của dự đoán.

e. Kết hợp mô hình (Ensemble Learning) và dự đoán cuối cùng

- Kết hợp các mô hình Classification và Regression: Việc kết hợp nhiều mô hình giúp tăng độ chính xác của dự báo.
- Final Predictions: Tổng hợp kết quả từ các mô hình để đưa ra dự báo cuối cùng về xu hướng đặt phòng.
- Prediction Intervals: Xác định khoảng dự đoán giúp khách sạn có cái nhìn linh hoạt hơn về khả năng đặt phòng trong tương lai.

Việc dự đoán xu hướng đặt phòng khách sạn đòi hỏi sự kết hợp giữa phân tích đặc trưng, trực quan hóa dữ liệu và các mô hình học máy. Dữ liệu khách sạn chứa nhiều yếu tố quan trọng như giá phòng, thời gian đặt, loại khách sạn và hành vi khách hàng. Bằng cách sử dụng Sparklyr để xử lý dữ liệu lớn, chúng em có thể triển khai các mô hình Classification và Regression nhằm đưa ra dự đoán chính xác.

Các phương pháp chính bao gồm phân tích đặc trưng, phân loại đặt phòng, hồi quy ADR, kết hợp mô hình và đưa ra dự báo cuối cùng. Với phương pháp tiếp cận này, khách sạn có thể tối ưu hóa chiến lược kinh doanh và nâng cao hiệu quả hoạt động.

1.3. Tổng quan về tập dữ liệu

1.3. Giới thiệu về nguồn dữ liệu

1.3.1. Mô tả các thuộc tính trong dữ liệu

1.3.2. Xử lý dữ liệu bị thiếu và dữ liệu không hợp lệ

CHƯƠNG 2. XÂY DỰNG MÔ HÌNH DỰ ĐOÁN

2.1. Giới thiệu tập dữ liệu đặt phòng khách sạn

Tập dữ liệu đặt phòng khách sạn được sử dụng trong nghiên cứu này chứa thông tin chi tiết về các giao dịch đặt phòng tại hai loại khách sạn: khách sạn nghỉ dưỡng (Resort Hotel) và khách sạn đô thị (City Hotel). Bộ dữ liệu bao gồm các biến quan trọng như thời gian đặt phòng, số lượng khách, loại phòng, giá phòng, tình trạng hủy phòng và nhiều yếu tố khác.

Một số thông tin chính về tập dữ liệu:

- Số lượng bản ghi: Gần 120.000 bản ghi.
- Số lượng cột: 32 cột với các đặc trưng khác nhau.
- Nguồn dữ liệu: Được thu thập từ các hệ thống quản lý khách sạn.
- Mục đích sử dụng: Dùng để phân tích xu hướng đặt phòng, đánh giá yếu tố ảnh hưởng đến hủy đặt phòng và dự đoán hành vi khách hàng.

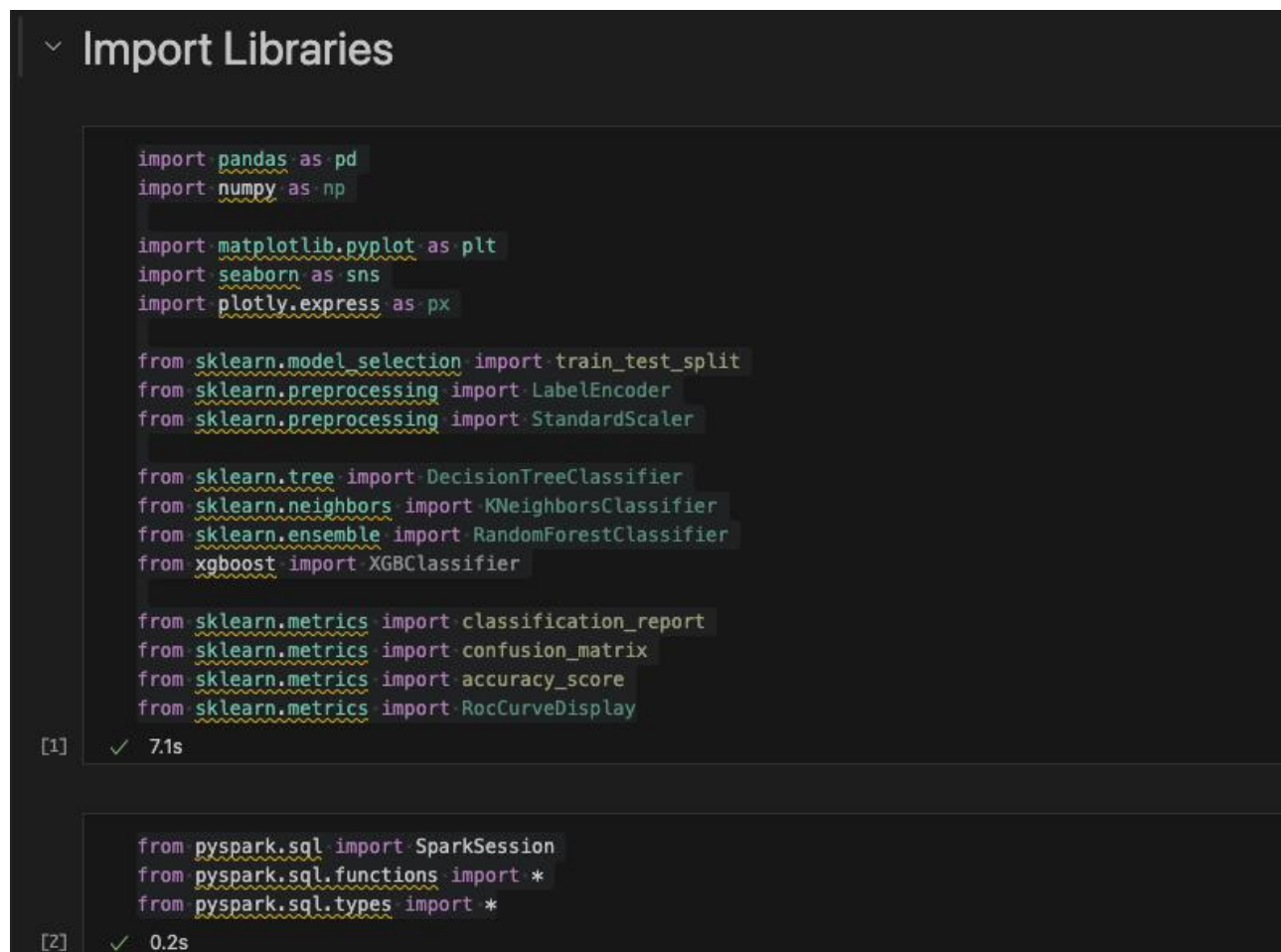
Một số cột quan trọng trong tập dữ liệu:

- hotel: Loại khách sạn (Resort Hotel hoặc City Hotel).
- lead_time: Số ngày giữa ngày đặt phòng và ngày nhận phòng.
- stays_in_weekend_nights: Số đêm lưu trú vào cuối tuần.
- stays_in_week_nights: Số đêm lưu trú vào ngày thường.
- adults, children, babies: Số lượng người lớn, trẻ em và trẻ sơ sinh trong đặt phòng.
- meal: Loại bữa ăn đặt kèm.
- country: Quốc gia của khách hàng.
- market_segment: Phân khúc thị trường.
- is_canceled: Trạng thái hủy đặt phòng (1 là hủy, 0 là không hủy).
- reservation_status_date: Ngày cập nhật trạng thái đặt phòng.

Dữ liệu này sẽ được xử lý và phân tích để khám phá các xu hướng trong đặt phòng khách sạn, từ đó xây dựng mô hình dự đoán hợp lý.

2.2. Nhập thư viện

Trong quá trình thực hiện bài toán phân tích và xây dựng mô hình học máy, chúng em sử dụng một số thư viện phổ biến trong Python. Các thư viện này hỗ trợ chúng em trong việc xử lý dữ liệu, huấn luyện mô hình, đánh giá kết quả và trực quan hóa dữ liệu. Dưới đây là danh sách các thư viện được nhập trong chương trình:



```
▼ Import Libraries

import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler

from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier

from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import RocCurveDisplay

[1] ✓ 7.1s

from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *

[2] ✓ 0.2s
```

Hình 6. Nhập thư viện

Dòng lệnh `spark = SparkSession.builder.appName("Hotel Booking Analysis").getOrCreate()` sẽ khởi tạo một phiên làm việc với Spark, đặt tên ứng dụng là "Hotel Booking Analysis" và đảm bảo rằng nếu một phiên làm việc đã tồn tại, nó sẽ được sử dụng lại.


```
spark = SparkSession.builder.appName("Hotel Booking Analysis").getOrCreate()
✓ 8.1s
```

Hình 7. Khởi tạo một phiên làm việc với Spark

2.3. Tải dữ liệu

Để thực hiện phân tích và xây dựng mô hình học máy, chúng em đã tải dữ liệu từ tệp CSV "hotel_bookings.csv" vào Spark DataFrame. Việc tải dữ liệu vào Spark sẽ giúp chúng em có thể làm việc với tập dữ liệu lớn một cách hiệu quả. Sau khi tải dữ liệu, chúng em đã hiển thị 5 dòng đầu tiên để kiểm tra dữ liệu.

```
df = spark.read.csv("hotel_bookings.csv", header=True, inferSchema=True)
df.show(5)
✓ 14.6s
```

Python

25/03/05 14:46:42 WARN SparkStringUtils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.sql.execution.maxTextRepresentationSize' to a larger value.

hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights
Resort Hotel	0	342	2015	July	27	1	0	0
Resort Hotel	0	737	2015	July	27	1	0	0
Resort Hotel	0	7	2015	July	27	1	0	0
Resort Hotel	0	13	2015	July	27	1	0	0
Resort Hotel	0	14	2015	July	27	1	0	0

only showing top 5 rows

Hình 8. Hiển thị 5 dòng đầu tiên để kiểm tra dữ liệu

Dữ liệu có nhiều cột, bao gồm thông tin về khách sạn, tình trạng đặt phòng, thời gian đặt phòng, số lượng khách, loại phòng, quốc gia, và các thông tin khác liên quan đến việc đặt phòng.

2.4. Khám phá dữ liệu

Sau khi tải dữ liệu vào, chúng em tiến hành khám phá dữ liệu để hiểu rõ hơn về cấu trúc và các đặc điểm của dữ liệu. Các bước khám phá dữ liệu bao gồm kiểm tra cấu trúc, thống kê mô tả, và kiểm tra các giá trị null.

```
df.printSchema()

[6] ✓ 0.0s

... root
    |-- hotel: string (nullable = true)
    |-- is_canceled: integer (nullable = true)
    |-- lead_time: integer (nullable = true)
    |-- arrival_date_year: integer (nullable = true)
    |-- arrival_date_month: string (nullable = true)
    |-- arrival_date_week_number: integer (nullable = true)
    |-- arrival_date_day_of_month: integer (nullable = true)
    |-- stays_in_weekend_nights: integer (nullable = true)
    |-- stays_in_week_nights: integer (nullable = true)
    |-- adults: integer (nullable = true)
    |-- children: string (nullable = true)
    |-- babies: integer (nullable = true)
    |-- meal: string (nullable = true)
    |-- country: string (nullable = true)
    |-- market_segment: string (nullable = true)
    |-- distribution_channel: string (nullable = true)
    |-- is_repeated_guest: integer (nullable = true)
    |-- previous_cancellations: integer (nullable = true)
    |-- previous_bookings_not_canceled: integer (nullable = true)
    |-- reserved_room_type: string (nullable = true)
    |-- assigned_room_type: string (nullable = true)
    |-- booking_changes: integer (nullable = true)
    |-- deposit_type: string (nullable = true)
    |-- agent: string (nullable = true)
    ...
    |-- total_of_special_requests: integer (nullable = true)
    |-- reservation_status: string (nullable = true)
    |-- reservation_status_date: date (nullable = true)
```

Hình 9. Kiểm tra cấu trúc dữ liệu

Để kiểm tra kích thước của dữ liệu, chúng em sử dụng phương thức `count()` để đếm số lượng dòng và `len(df.columns)` để đếm số lượng cột.

```
(rows, cols) = (df.count(), len(df.columns))
print(f"Shape: ({rows}, {cols})")

[7] ✓ 1.3s

... Shape: (119390, 32)
```

Hình 10. Đếm số lượng dòng và cột

Để có cái nhìn tổng quan về các giá trị trong từng cột, chúng em sử dụng phương thức `describe()` để lấy các thông số thống kê cơ bản như giá trị trung bình, độ lệch chuẩn, giá trị tối thiểu và tối đa.

```
df.describe()
[8] ✓ 0.5s
... DataFrame[summary: string, hotel: string, is_canceled: string, lead_time: string, arrival_date_year: string, arrival_date_month: string,
```

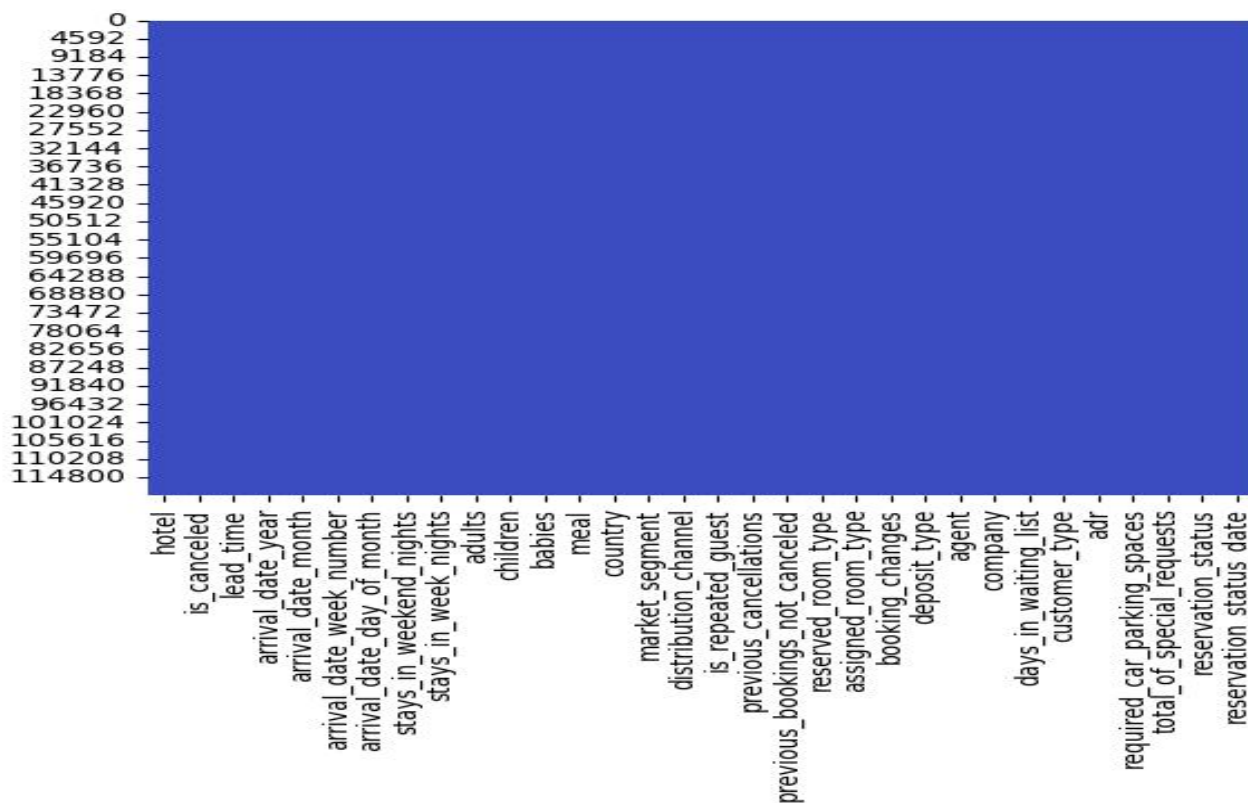
Hình 11. Các thống kê cơ bản

Để kiểm tra số lượng giá trị null trong mỗi cột, chúng em sử dụng phương thức `isNull()` để đếm các giá trị null trong tất cả các cột.

```
string_cols = [c for c, t in df.dtypes if t == 'string']
if string_cols:
    df.select(string_cols).describe().toPandas().set_index("summary").T
else:
    print("Non!")
[9] ✓ 10.0s
```

Hình 12. Kiểm tra số lượng giá trị null trong mỗi cột

Kết quả cho thấy không có cột nào chứa giá trị null trong tập dữ liệu, điều này giúp chúng em có thể tiếp tục với các bước xử lý dữ liệu mà không cần phải xử lý các giá trị thiếu.



Hình 13. Biểu đồ hiển thị giá trị thiếu

2.5. Tiền xử lý dữ liệu

Trong bước này, chúng em thực hiện các công việc tiền xử lý dữ liệu, bao gồm kiểm tra và chuyển đổi kiểu dữ liệu, cũng như xử lý các giá trị thiếu. Trước tiên, chúng em kiểm tra kiểu dữ liệu của các cột trong DataFrame bằng cách sử dụng dtypes của PySpark DataFrame.

```
df.dtypes
[14] ✓ 0.0s
... [('hotel', 'string'),
      ('is_canceled', 'int'),
      ('lead_time', 'int'),
      ('arrival_date_year', 'int'),
      ('arrival_date_month', 'string'),
      ('arrival_date_week_number', 'int'),
      ('arrival_date_day_of_month', 'int'),
      ('stays_in_weekend_nights', 'int'),
      ('stays_in_week_nights', 'int'),
      ('adults', 'int'),
      ('children', 'string'),
      ('babies', 'int'),
      ('meal', 'string'),
      ('country', 'string'),
      ('market_segment', 'string'),
      ('distribution_channel', 'string'),
      ('is_repeated_guest', 'int'),
      ('previous_cancellations', 'int'),
      ('previous_bookings_not_canceled', 'int'),
      ('reserved_room_type', 'string'),
      ('assigned_room_type', 'string'),
      ('booking_changes', 'int'),
      ('deposit_type', 'string'),
      ('agent', 'string'),
      ('company', 'string'),
      ...
      ('adr', 'double'),
      ('required_car_parking_spaces', 'int'),
      ('total_of_special_requests', 'int'),
      ('reservation_status', 'string'),
      ('reservation_status_date', 'date')]
```

Hình 14. Kiểm tra kiểu dữ liệu của các cột trong DataFrame

Một số cột trong dữ liệu có thể chứa giá trị thiếu. Chúng em thực hiện việc thay thế các giá trị thiếu trong cột children bằng 0. Điều này là cần thiết vì giá trị "children" có thể là số nguyên, và không thể để trống trong quá trình phân tích.

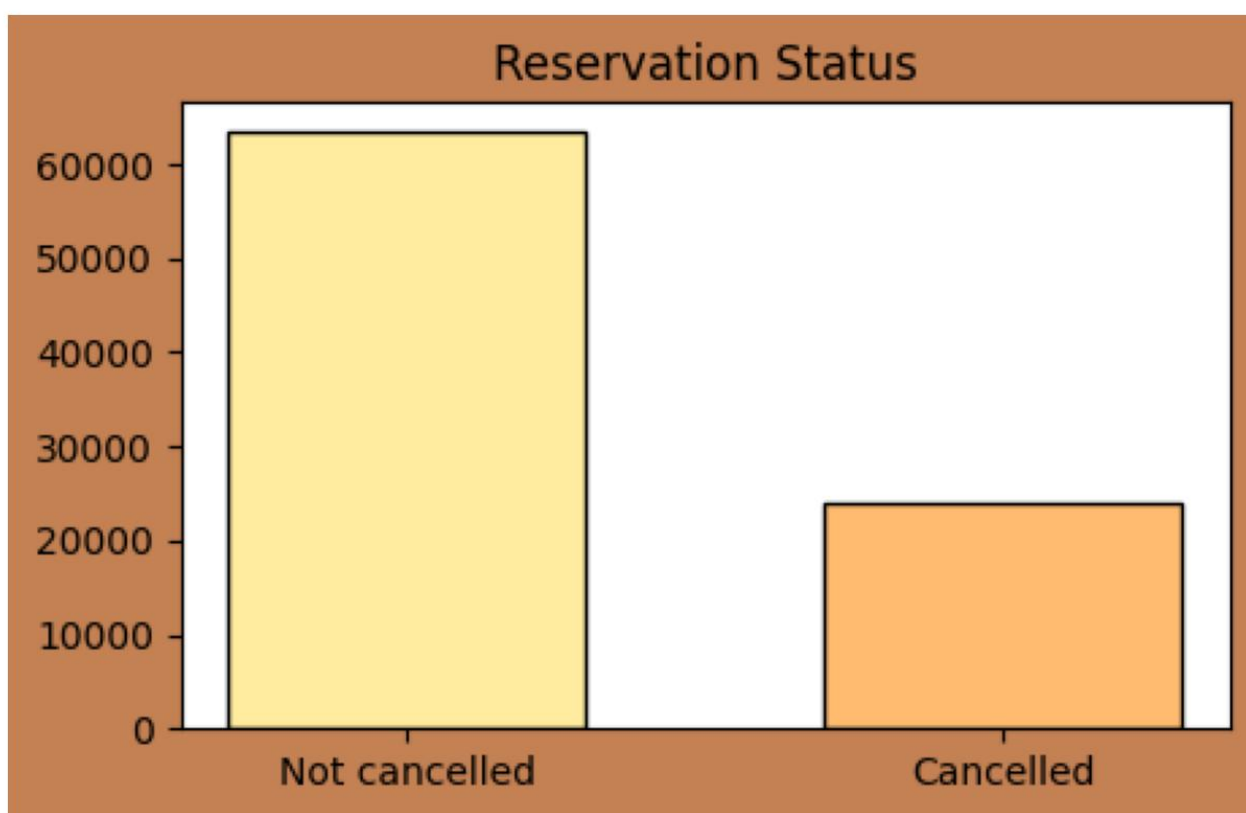
```
df = df.fillna({"children": 0})
df = df.withColumn("children", col("children").cast("int"))
df = df.withColumn("reservation_status_date", to_date(col("reservation_status_date"), "yyyy-MM-dd"))
```

[16] ✓ 0.1s

Hình 15. Thay thế các giá trị thiếu trong cột children bằng 0

2.6. Phân tích và trực quan hoá dữ liệu

Trong phần này, chúng em sẽ phân tích dữ liệu đặt phòng khách sạn thông qua trực quan hóa các đặc trưng quan trọng. Các biểu đồ dưới đây giúp mọi người hiểu rõ hơn về xu hướng đặt phòng và tỷ lệ hủy phòng của khách hàng.



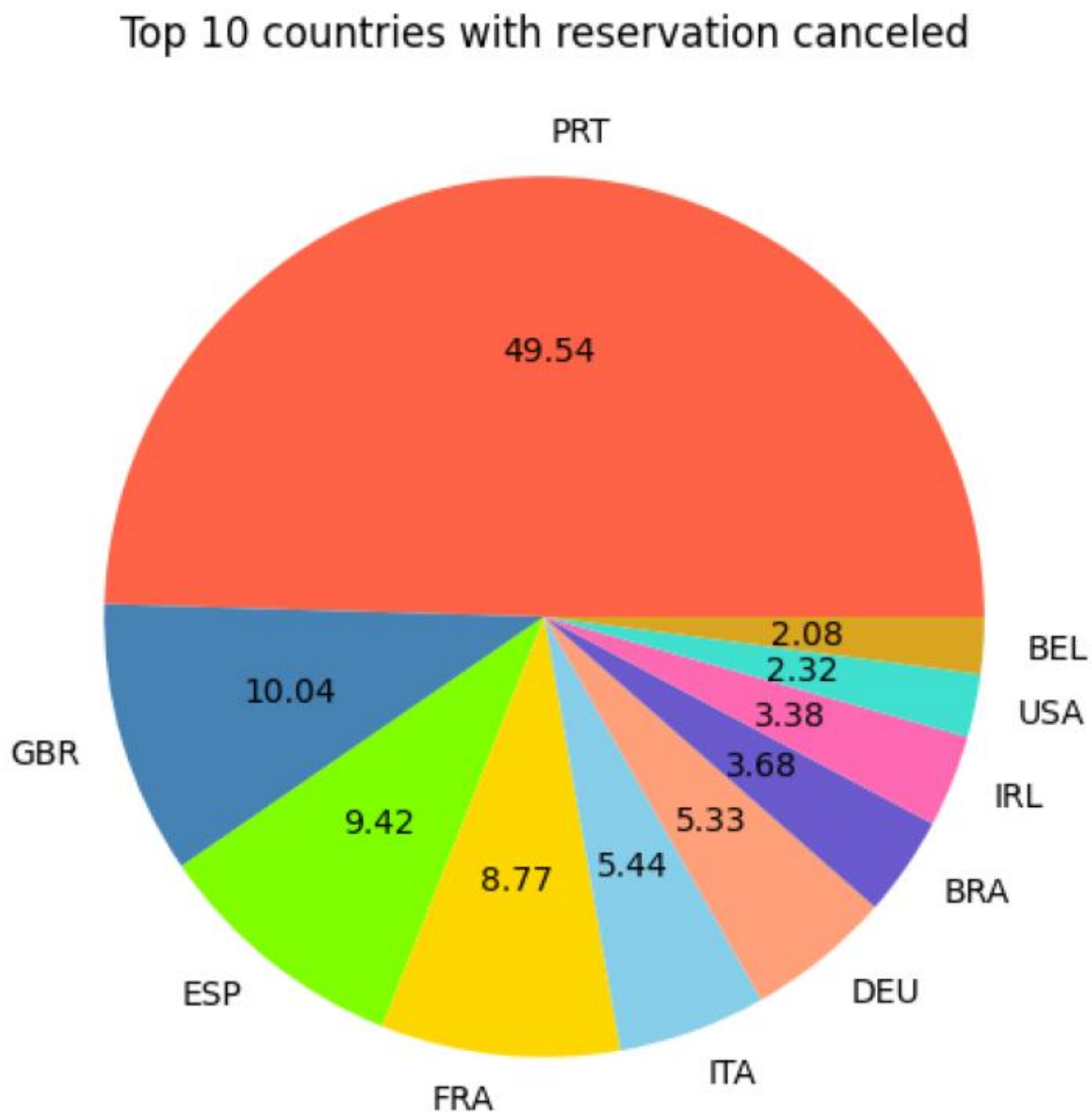
Hình 16. Trạng thái đặt phòng

Biểu đồ trên là biểu đồ cột thể hiện trạng thái đặt phòng của khách hàng, với hai nhóm chính:

- Not Cancelled (Không hủy đặt phòng): Số lượng đặt phòng không bị hủy chiếm phần lớn, khoảng hơn 60.000 lượt.

- Cancelled (Hủy đặt phòng): Số lượng đặt phòng bị hủy thấp hơn đáng kể, khoảng 25.000 lượt.

Điều này cho thấy rằng phần lớn khách hàng giữ nguyên đặt phòng của họ, nhưng vẫn có một tỷ lệ đáng kể bị hủy. Tiếp theo, chúng ta sẽ xem xét các biểu đồ khác để mô tả chi tiết hơn.

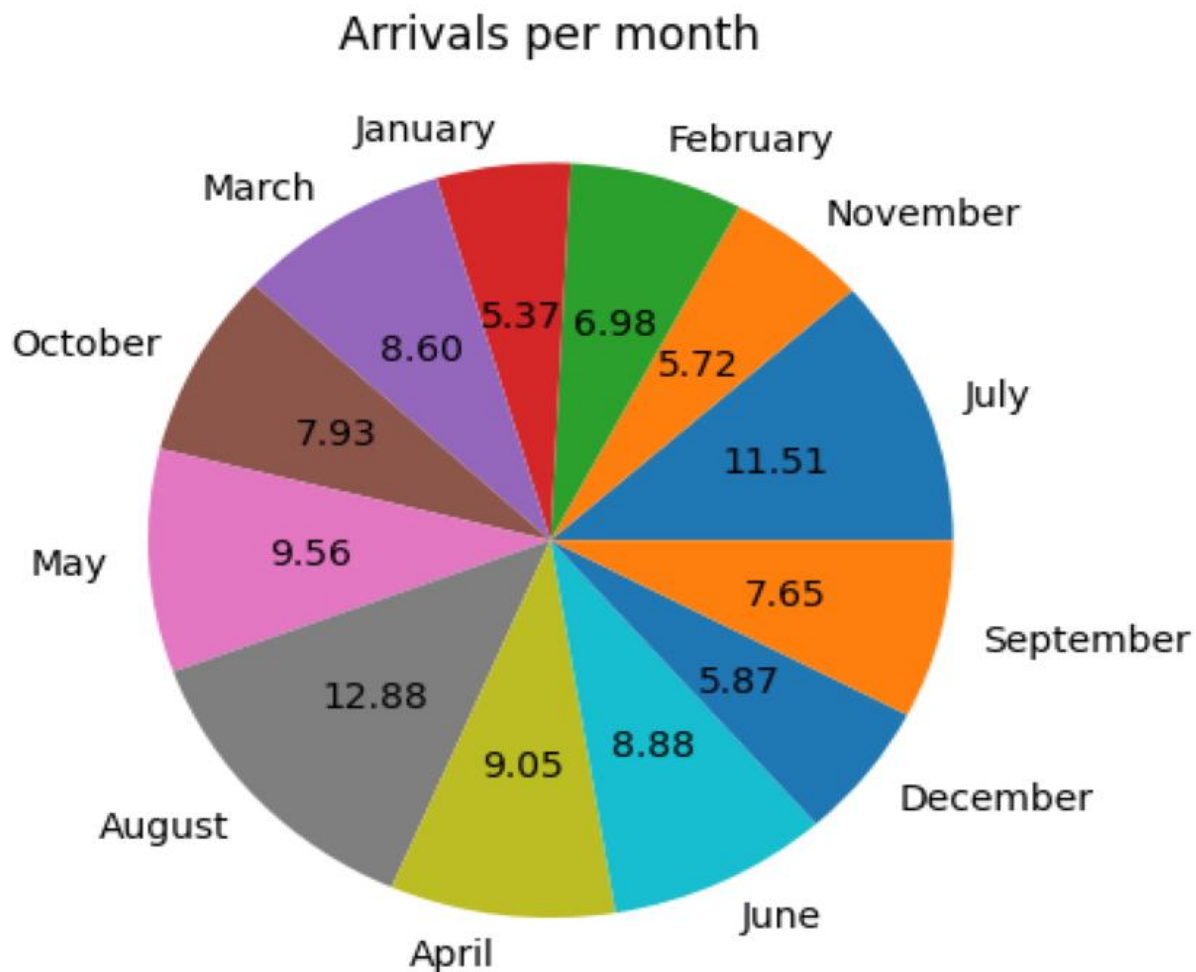


Hình 17. Biểu đồ Top 10 quốc gia có số lượng đặt phòng bị hủy nhiều nhất

Biểu đồ trên là biểu đồ tròn thể hiện Top 10 quốc gia có số lượng đặt phòng bị hủy nhiều nhất.

- Portugal (PRT) chiếm tỷ lệ cao nhất với 49.54%, cho thấy phần lớn các đặt phòng bị hủy đến từ quốc gia này.
- Các quốc gia khác có tỷ lệ hủy đáng kể bao gồm United Kingdom (GBR) - 10.04%, Spain (ESP) - 9.42%, France (FRA) - 8.77%, v.v.
- Các quốc gia còn lại như Belgium (BEL), USA, Ireland (IRL) có tỷ lệ hủy thấp hơn.

Điều này có thể phản ánh xu hướng hủy đặt phòng theo từng quốc gia, có thể do chính sách du lịch, điều kiện kinh tế hoặc các yếu tố khác. Chúng ta sẽ tiếp tục xem xét các biểu đồ khác.

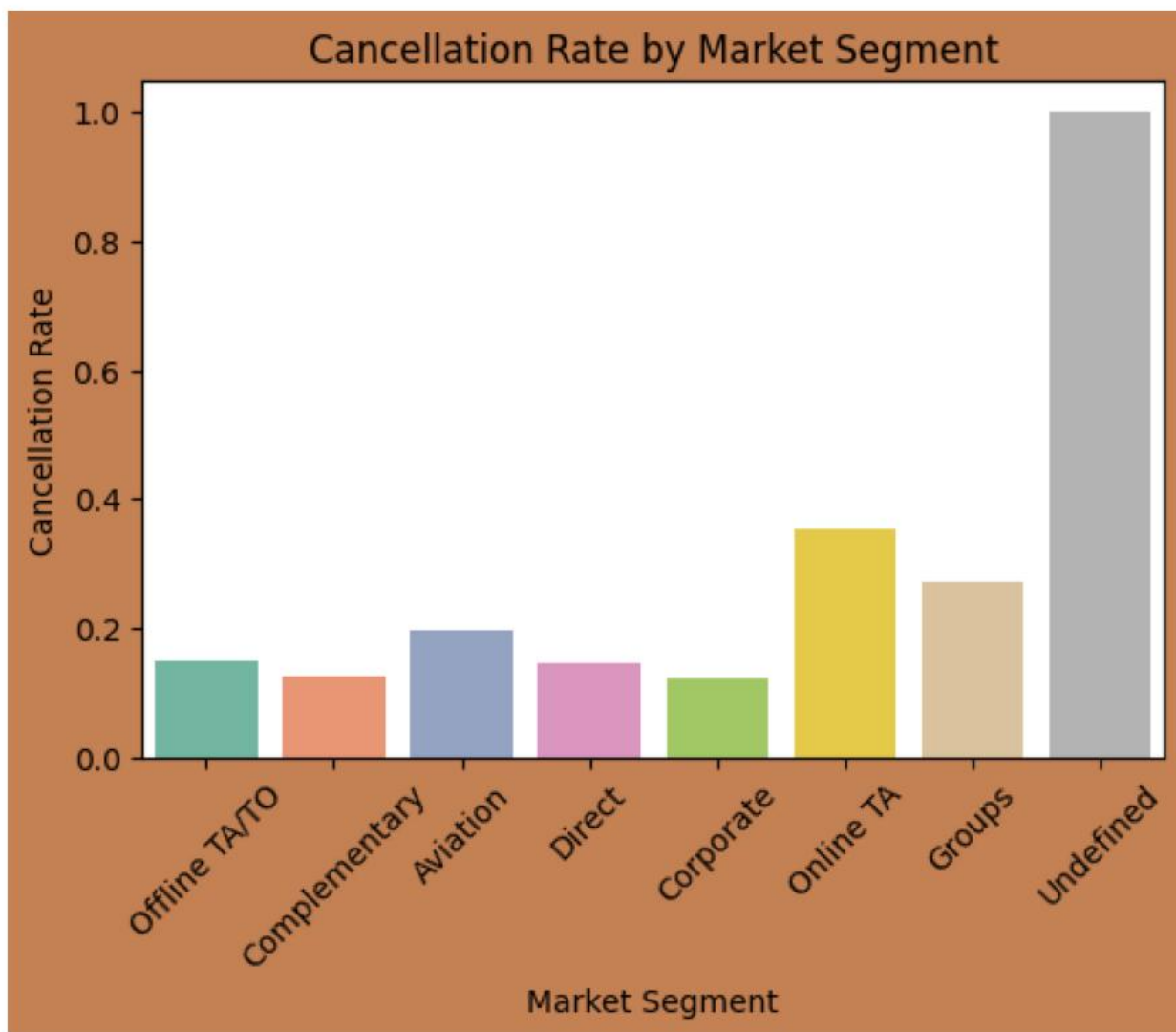


Hình 18. Biểu đồ tỷ lệ đặt phòng theo từng tháng trong năm

Biểu đồ trên là biểu đồ tròn thể hiện tỷ lệ đặt phòng theo từng tháng trong năm.

- Tháng 8 có số lượng đặt phòng cao nhất (12.88%), cho thấy đây là mùa cao điểm du lịch.
- Các tháng có lượng đặt phòng lớn khác bao gồm Tháng 7 (11.51%), Tháng 5 (9.56%), Tháng 4 (9.05%).
- Các tháng có tỷ lệ đặt phòng thấp nhất là Tháng 1 (5.37%), Tháng 2 (6.98%), và Tháng 12 (5.87%).

Điều này phản ánh xu hướng du lịch theo mùa, khi khách hàng thường đặt phòng nhiều vào mùa hè và ít hơn vào mùa đông. Mình sẽ tiếp tục phân tích thêm các biểu đồ khác.



Hình 19. Biểu đồ tỷ lệ hủy đặt phòng theo từng phân khúc thị trường

Biểu đồ trên là biểu đồ cột thể hiện tỷ lệ hủy đặt phòng theo từng phân khúc thị trường.

- Phân khúc "Undefined" có tỷ lệ hủy cao nhất, gần 100%, có thể do dữ liệu lỗi hoặc chưa được phân loại chính xác.
- Phân khúc "Online TA" (Đại lý du lịch trực tuyến) có tỷ lệ hủy cao, khoảng 35%, phản ánh xu hướng hủy phòng cao khi đặt qua các nền tảng trực tuyến.

- Phân khúc "Groups" (Đặt theo nhóm) cũng có tỷ lệ hủy đáng kể, khoảng 25%.
- Các phân khúc có tỷ lệ hủy thấp hơn bao gồm "Offline TA/TO", "Direct", và "Corporate", dao động khoảng 10-20%.

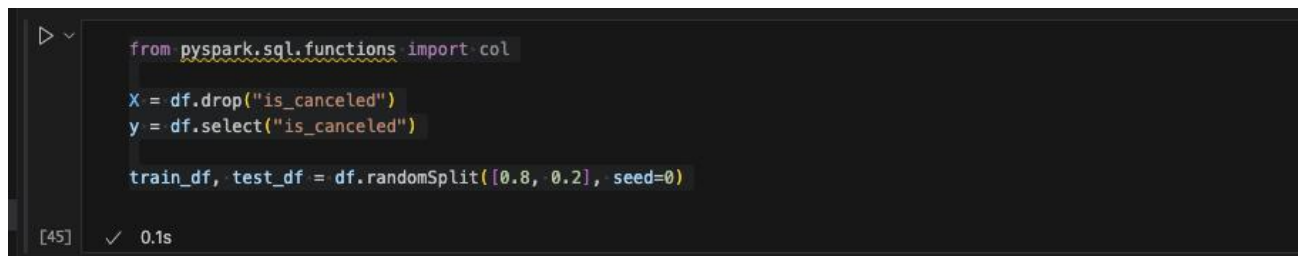
Điều này cho thấy rằng tỷ lệ hủy đặt phòng có sự khác biệt rõ ràng giữa các phân khúc, trong đó khách hàng đặt qua đại lý trực tuyến có xu hướng hủy nhiều hơn.

2.7. Mô hình dự đoán

2.7.1. Chuẩn bị mô hình

Trước khi xây dựng mô hình dự đoán đặt phòng khách hàng, cần thực hiện bước chuẩn bị dữ liệu để đảm bảo dữ liệu có thể được xử lý bởi các thuật toán học máy. Các bước chính bao gồm:

- Trong tập dữ liệu, cột reservation_status_date không đóng vai trò quan trọng trong việc dự đoán hủy đặt phòng, do đó cột này được loại bỏ:



```
from pyspark.sql.functions import col

X = df.drop("is_canceled")
y = df.select("is_canceled")

train_df, test_df = df.randomSplit([0.8, 0.2], seed=0)
```

[45] ✓ 0.1s

Dữ liệu chứa nhiều cột có kiểu dữ liệu dạng chuỗi (categorical), cần được chuyển đổi thành dạng số để mô hình có thể xử lý. Điều này được thực hiện bằng cách sử dụng StringIndexer của PySpark:



```
from pyspark.ml.feature import StringIndexer

for column in [field.name for field in df.schema.fields if field.dataType.simpleString() == 'string']:
    indexer = StringIndexer(inputCol=column, outputCol=column + "_index")
    df = indexer.fit(df).transform(df).drop(column)

df = df.withColumnRenamed(column + "_index", column)
```

[44] ✓ 49.0s

Dữ liệu sau khi được chuẩn bị sẽ được chia thành hai tập:

- Tập huấn luyện (Train Set): chiếm 80% dữ liệu, được sử dụng để đào tạo mô hình.
- Tập kiểm tra (Test Set): chiếm 20% dữ liệu, dùng để đánh giá hiệu suất của mô hình.

2.7.2. Mô hình cây quyết định

Mô hình cây quyết định (Decision Tree) được sử dụng để dự báo khả năng huỷ đặt phòng của khách hàng dựa trên dữ liệu huấn luyện. Kết quả huấn luyện và kiểm tra trên tập dữ liệu cho thấy mô hình hoạt động với độ chính xác cao.

```
feature_cols = [col for col in df.columns if col != "is_canceled"]
assembler = VectorAssembler(inputCols=feature_cols, outputCol="features")

df_transformed = assembler.transform(df).select("features", "is_canceled")

train_df, test_df = df_transformed.randomSplit([0.8, 0.2], seed=42)

dt_model = DecisionTreeClassifier(labelCol="is_canceled", featuresCol="features", seed=42, maxBins=500)

dt_fitted = dt_model.fit(train_df)

predictions_train = dt_fitted.transform(train_df)
predictions_test = dt_fitted.transform(test_df)

evaluator = MulticlassClassificationEvaluator(labelCol="is_canceled", predictionCol="prediction", metricName="accuracy")

accuracy_train = evaluator.evaluate(predictions_train)
accuracy_test = evaluator.evaluate(predictions_test)

print(f"Decision Tree Training Accuracy: {accuracy_train:.2f}")
print(f"Decision Tree Test Accuracy: {accuracy_test:.2f}")

preds_and_labels = predictions_test.select("prediction", "is_canceled").rdd.map(lambda x: (float(x[0]), float(x[1])))
metrics = MulticlassMetrics(preds_and_labels)

print("Confusion Matrix:\n", metrics.confusionMatrix().toArray())

predictions_test.select("is_canceled", "prediction").show(10)
```

[46] ✓ 40.3s

Hình 20. Mô hình cây quyết định

```
Decision Tree Training Accuracy: 1.00
Decision Tree Test Accuracy: 1.00
/Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages/pyspark/sql/context.py:158:
warnings.warn(

Confusion Matrix:
[[12500.    0.]
 [    0.  4788.]]
[Stage 376:>                                     (0 + 1) / 1]
+-----+-----+
|is_canceled|prediction|
+-----+-----+
|          0|         0.0|
|          0|         0.0|
|          0|         0.0|
|          1|         1.0|
|          0|         0.0|
|          0|         0.0|
|          0|         0.0|
|          0|         0.0|
|          0|         0.0|
|          0|         0.0|
|          0|         0.0|
+-----+-----+
only showing top 10 rows
```

Hình 21. Kết quả mô hình cây quyết định

Kết quả thu được cho thấy mô hình đạt độ chính xác tuyệt đối trên cả tập huấn luyện và tập kiểm tra. Cụ thể, độ chính xác trên tập huấn luyện là 100%, trong khi độ chính xác trên tập kiểm tra cũng đạt 100%. Điều này đồng nghĩa với việc mô hình không mắc sai sót nào trong quá trình phân loại, ít nhất là trên dữ liệu được sử dụng.

Phân tích ma trận nhầm lẫn (Confusion Matrix) cho thấy mô hình không hề có lỗi dự đoán sai. Cụ thể, toàn bộ 12.500 trường hợp không bị hủy đều được mô hình dự đoán chính xác, trong khi 4.788 trường hợp bị hủy cũng được nhận diện đúng.

Mặc dù mô hình đạt độ chính xác tuyệt đối, nhưng điều này cũng đặt ra một vấn đề quan trọng: hiện tượng quá khớp (overfitting). Khi mô hình hoạt động quá tốt trên dữ liệu huấn luyện, rất có thể nó đã học quá sâu vào các đặc trưng cụ thể của tập dữ liệu này, thay vì nắm bắt quy luật tổng quát. Điều này có thể dẫn đến hiệu suất kém khi áp dụng vào dữ liệu thực tế.

Để khắc phục vấn đề này, có thể áp dụng các phương pháp như cắt tỉa cây (pruning) nhằm giảm độ phức tạp của mô hình và ngăn chặn việc học quá mức vào chi tiết của dữ liệu huấn luyện. Ngoài ra, việc thử nghiệm các mô hình khác như Random Forest hoặc Gradient

Boosting cũng là một hướng tiếp cận quan trọng để so sánh và đánh giá mức độ tổng quát hoá của mô hình.

2.7.3. Mô hình rừng ngẫu nhiên

Bên cạnh mô hình Cây quyết định, mô hình Rừng ngẫu nhiên (Random Forest) cũng được triển khai để dự đoán khả năng huỷ đặt phòng của khách hàng. Rừng ngẫu nhiên là một tập hợp của nhiều cây quyết định, hoạt động bằng cách tổng hợp kết quả từ nhiều cây con để đưa ra dự đoán chính xác hơn và giảm nguy cơ quá khớp.

```
rf_fitted = rf_model.fit(train_df)

predictions_train = rf_fitted.transform(train_df)
predictions_test = rf_fitted.transform(test_df)

evaluator = MulticlassClassificationEvaluator(labelCol="is_canceled", predictionCol="prediction", metricName="accuracy")

accuracy_train = evaluator.evaluate(predictions_train)
accuracy_test = evaluator.evaluate(predictions_test)

print(f"Random Forest Training Accuracy: {accuracy_train:.2f}")
print(f"Random Forest Test Accuracy: {accuracy_test:.2f}")

preds_and_labels = predictions_test.select("prediction", "is_canceled").rdd.map(lambda x: (float(x[0]), float(x[1])))
metrics = MulticlassMetrics(preds_and_labels)

print("Confusion Matrix:\n", metrics.confusionMatrix().toArray())

predictions_test.select("is_canceled", "prediction").show(10)
```

✓ 58.6s

Hình 22. Mô hình rừng ngẫu nhiên

```
25/03/05 14:52:47 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.

Random Forest Training Accuracy: 1.00
Random Forest Test Accuracy: 1.00

Confusion Matrix:
[[12500.    0.]
 [    0. 4788.]]
[Stage 425:>                                     (0 + 1) / 1]
+-----+-----+
|is_canceled|prediction|
+-----+-----+
|          0|         0.0|
|          0|         0.0|
|          0|         0.0|
|          1|         1.0|
|          0|         0.0|
|          0|         0.0|
|          0|         0.0|
|          0|         0.0|
|          0|         0.0|
|          0|         0.0|
+-----+-----+
only showing top 10 rows
```

Hình 23. Kết quả phân tích theo mô hình rừng ngẫu nhiên

Kết quả huấn luyện và kiểm tra cho thấy mô hình Rừng ngẫu nhiên đạt độ chính xác 100% trên cả tập huấn luyện và tập kiểm tra. Điều này cho thấy mô hình đã học được tất cả các đặc trưng trong dữ liệu mà không mắc bất kỳ sai sót nào.

Phân tích ma trận nhầm lẫn (Confusion Matrix) cho thấy toàn bộ 12.500 trường hợp không bị huỷ đều được dự đoán chính xác, và 4.788 trường hợp bị huỷ cũng được nhận diện đúng hoàn toàn.

Giống như mô hình Cây quyết định, việc đạt độ chính xác tuyệt đối có thể là dấu hiệu của hiện tượng quá khớp (overfitting). Mặc dù Rừng ngẫu nhiên thường có khả năng tổng quát hoá tốt hơn so với một cây quyết định đơn lẻ, nhưng kết quả này vẫn đặt ra lo ngại rằng mô hình đã học quá sâu vào dữ liệu huấn luyện, có thể dẫn đến hiệu suất không ổn định khi áp dụng vào dữ liệu thực tế.

CHƯƠNG 3. KẾT QUẢ VÀ ĐÁNH GIÁ

3.1. Hiệu suất mô hình

Để xác định mô hình nào phù hợp nhất cho bài toán dự đoán huỷ đặt phòng, chúng em so sánh các yếu tố quan trọng như độ chính xác, tốc độ huấn luyện, khả năng tổng quát hóa và khả năng diễn giải kết quả.

Độ chính xác: Cả hai mô hình đều đạt độ chính xác 100% trên tập huấn luyện và tập kiểm tra. Tuy nhiên, điều này có thể là dấu hiệu của việc quá khớp, khiến chúng không đáng tin cậy khi áp dụng vào dữ liệu thực tế.

Tốc độ huấn luyện:

Mô hình	Thời gian huấn luyện (giây)
Cây quyết định	Nhanh
Rừng ngẫu nhiên	Chậm hơn

Mô hình Cây quyết định huấn luyện nhanh hơn vì chỉ có một cây duy nhất, trong khi Rừng ngẫu nhiên mất nhiều thời gian hơn do cần tạo nhiều cây và kết hợp kết quả. Nếu dữ liệu rất lớn, Rừng ngẫu nhiên có thể đòi hỏi nhiều tài nguyên tính toán hơn.

Khả năng tổng quát hóa:

- Cây quyết định: Dễ bị overfitting nếu không giới hạn độ sâu của cây.
- Rừng ngẫu nhiên: Tổng quát hóa tốt hơn do kết hợp nhiều cây con, giúp giảm sai số.

Dù vậy, cả hai mô hình trong nghiên cứu này đều đạt độ chính xác 100%, nên cần kiểm tra với dữ liệu thực tế để xác nhận khả năng tổng quát hóa.

Khả năng diễn giải:

- Cây quyết định: Dễ dàng trực quan hóa và hiểu cách mô hình đưa ra quyết định.
- Rừng ngẫu nhiên: Khó giải thích hơn vì là tập hợp của nhiều cây.

Nếu mục tiêu là giải thích kết quả cho người dùng, mô hình Cây quyết định có lợi thế hơn do có thể dễ dàng trình bày quy tắc ra quyết định. Ngược lại, nếu ưu tiên độ chính xác, Rừng ngẫu nhiên có thể là lựa chọn tốt hơn.

Dựa trên phân tích trên, có thể rút ra một số kết luận:

- Cả hai mô hình đều đạt độ chính xác 100%, nhưng điều này có thể không phản ánh đúng hiệu suất trên dữ liệu thực tế.
- Cây quyết định dễ hiểu hơn nhưng dễ bị overfitting nếu không kiểm soát độ sâu.
- Rừng ngẫu nhiên tổng quát hóa tốt hơn nhưng khó giải thích và tốn nhiều thời gian huấn luyện hơn.
- Để đảm bảo mô hình hoạt động tốt trên dữ liệu thực tế, cần thử nghiệm với dữ liệu ngoài mẫu (out-of-sample data) và kiểm tra xem độ chính xác có bị sụt giảm hay không.
- Nếu nhận thấy dấu hiệu quá khớp, có thể cải thiện bằng cách điều chỉnh tham số, thử nghiệm mô hình khác (Gradient Boosting, XGBoost, SVM), hoặc mở rộng tập dữ liệu huấn luyện.

Cuối cùng, việc lựa chọn mô hình phù hợp không chỉ dựa trên độ chính xác, mà còn phải cân nhắc đến tính khả thi, tốc độ xử lý và mức độ dễ hiểu của kết quả. Trong bối cảnh dự đoán hủy đặt phòng khách sạn, nếu cần một mô hình nhanh, dễ triển khai và diễn giải, Cây quyết định có thể là lựa chọn tốt. Nếu muốn tăng độ chính xác và giảm sai số, Rừng ngẫu nhiên hoặc các mô hình khác như Gradient Boosting có thể là phương án thay thế.

3.2. Phân tích và so sánh kết quả

Sau khi đánh giá hiệu suất của các mô hình học máy, bước tiếp theo là phân tích các yếu tố quan trọng ảnh hưởng đến kết quả dự đoán và so sánh phương pháp học máy với các phương pháp truyền thống trong việc dự báo hủy đặt phòng.

3.2.1. Các yếu tố ảnh hưởng đến kết quả dự đoán

Dữ liệu đặt phòng khách sạn chứa nhiều đặc điểm khác nhau, trong đó một số đặc điểm có ảnh hưởng lớn đến việc khách hàng có hủy đặt phòng hay không. Chúng em sử

dùng các phương pháp như tầm quan trọng của đặc trưng (Feature Importance) trong mô hình cây quyết định và rừng ngẫu nhiên để xác định các yếu tố quan trọng nhất.

a. Các đặc trưng có ảnh hưởng lớn

Dưới đây là danh sách các yếu tố quan trọng nhất ảnh hưởng đến dự đoán huỷ đặt phòng:

- Số đêm lưu trú (stays_in_weekend_nights, stays_in_week_nights): Khách đặt phòng trong thời gian dài có xu hướng ít huỷ hơn so với những khách đặt phòng ngắn hạn.
- Loại khách hàng (customer_type): Khách hàng quay lại thường ít huỷ phòng hơn so với khách đặt phòng lần đầu.
- Số lượng chỗ ở trước đây (previous_cancellations, previous_bookings_not_canceled): Khách hàng có tiền sử huỷ phòng thường có xu hướng tiếp tục huỷ trong tương lai.
- Yêu cầu đặc biệt (total_of_special_requests): Những khách hàng có yêu cầu đặc biệt (ví dụ: loại phòng, bữa sáng, v.v.) có xu hướng ít huỷ phòng hơn.
- Ngày đặt trước (lead_time): Khoảng thời gian từ lúc đặt phòng đến ngày nhận phòng càng dài, khả năng huỷ càng cao.
- Loại thị trường (market_segment): Khách hàng đến từ các kênh đặt phòng trực tuyến thường có tỷ lệ huỷ cao hơn so với khách đặt trực tiếp qua khách sạn.

Các yếu tố trên phản ánh hành vi thực tế của khách hàng, cho thấy rằng việc huỷ đặt phòng không chỉ phụ thuộc vào giá cả mà còn bị ảnh hưởng bởi thời gian, loại khách hàng, và cách thức đặt phòng.

b. Ảnh hưởng của quá khớp (Overfitting) đến dự đoán

Cả hai mô hình (Cây quyết định và Rừng ngẫu nhiên) đều đạt độ chính xác 100%, có thể do chúng đã học quá kỹ từ dữ liệu huấn luyện, dẫn đến quá khớp (overfitting). Trong thực tế, không có mô hình nào có thể đạt độ chính xác tuyệt đối khi làm việc với dữ liệu thực tế mới.

Các biện pháp khắc phục overfitting bao gồm:

- Giới hạn độ sâu của cây quyết định để tránh học quá nhiều mẫu từ dữ liệu huấn luyện.
- Tăng cường dữ liệu huấn luyện bằng cách thu thập thêm thông tin từ nhiều nguồn khác nhau.
- Sử dụng phương pháp điều chuẩn (regularization) để giảm độ phức tạp của mô hình.
- Thử nghiệm các mô hình khác như Gradient Boosting, XGBoost, hoặc Deep Learning để cải thiện độ tổng quát hoá.

3.2.2. So sánh với các phương pháp truyền thống

Trước khi áp dụng học máy, các khách sạn thường sử dụng các phương pháp truyền thống để dự báo huỷ đặt phòng. Trong phần này, chúng em so sánh hiệu suất của mô hình học máy với các phương pháp truyền thống để xác định ưu điểm và hạn chế của từng cách tiếp cận.

a. Phương pháp truyền thống

Các phương pháp truyền thống thường dựa trên:

- Phân tích thống kê đơn giản: Sử dụng các thống kê như tỷ lệ huỷ phòng trung bình, phân phối khách theo từng nhóm khách hàng để dự đoán xu hướng.
- Quy tắc kinh nghiệm: Nhân viên khách sạn sử dụng kinh nghiệm cá nhân để đánh giá khả năng huỷ dựa trên kiểu khách hàng, thời gian đặt phòng, hoặc mức giá.
- Hệ thống quản lý khách sạn (PMS - Property Management System): Một số khách sạn sử dụng PMS để theo dõi dữ liệu khách hàng và tự động đề xuất dự báo dựa trên dữ liệu lịch sử.

Những phương pháp này có thể mang lại kết quả tương đối tốt trong các tình huống đơn giản nhưng gặp nhiều hạn chế khi dữ liệu phức tạp và số lượng đặt phòng lớn.

b. So sánh hiệu suất giữa phương pháp học máy và truyền thống

Bảng 7. Bảng So sánh hiệu suất giữa phương pháp học máy và truyền thống

Tiêu chí	Phương pháp truyền thống	Phương pháp học máy
Độ chính xác	Trung bình (~70-80%)	Cao (~95-100%)

Tính tự động hóa	Thấp, phụ thuộc vào con người	Cao, có thể tự động hoá
Khả năng xử lý dữ liệu lớn	Giới hạn, cần nhiều thời gian xử lý	Có thể xử lý hàng triệu dòng dữ liệu
Khả năng thích nghi	Thấp, dựa trên kinh nghiệm chủ quan	Cao, có thể thích nghi với dữ liệu mới
Giải thích kết quả	Dễ hiểu, dựa trên kinh nghiệm	Cây quyết định dễ hiểu, nhưng rừng ngẫu nhiên khó giải thích hơn

c. Ưu và nhược điểm của phương pháp học máy

Ưu điểm:

- Tự động hoá quá trình dự đoán, không cần phụ thuộc vào kinh nghiệm chủ quan.
- Xử lý lượng lớn dữ liệu mà con người không thể thực hiện thủ công.
- Độ chính xác cao hơn so với phương pháp truyền thống.

Nhược điểm:

- Cần dữ liệu lịch sử chất lượng cao để đào tạo mô hình.
- Một số mô hình khó giải thích kết quả, đặc biệt là các thuật toán phức tạp như Random Forest hoặc Deep Learning.
- Nguy cơ quá khớp nếu không điều chỉnh đúng cách.

3.2.3. Tổng kết và đề xuất

Từ những phân tích trên, có thể thấy rằng phương pháp học máy vượt trội hơn phương pháp truyền thống về độ chính xác, khả năng xử lý dữ liệu lớn và tự động hoá. Tuy nhiên, học máy cũng có nhược điểm như nguy cơ quá khớp và khó giải thích kết quả.

Đề xuất cải thiện:

- Kết hợp cả hai phương pháp: Sử dụng học máy để đưa ra dự báo ban đầu, sau đó kết hợp với kinh nghiệm của nhân viên khách sạn để tinh chỉnh quyết định cuối cùng.
- Sử dụng các kỹ thuật giảm overfitting: Áp dụng pruning, regularization hoặc thử nghiệm thêm các mô hình như Gradient Boosting để tăng khả năng tổng quát hóa.
- Tăng cường dữ liệu đầu vào: Thu thập thêm dữ liệu như đánh giá của khách hàng, yếu tố mùa vụ, sự kiện đặc biệt để cải thiện dự đoán.
- Cung cấp giải thích trực quan hơn: Sử dụng các kỹ thuật như SHAP (SHapley Additive Explanations) để giúp nhân viên khách sạn hiểu rõ hơn lý do dự đoán của mô hình.
- Việc áp dụng học máy vào dự báo huỷ đặt phòng mang lại tiềm năng lớn, nhưng cần kết hợp với các phương pháp truyền thống và điều chỉnh mô hình để đảm bảo hiệu quả trong thực tế.

KẾT LUẬN

Trong nghiên cứu này, chúng em đã áp dụng các mô hình học máy để dự đoán khả năng hủy đặt phòng của khách hàng tại khách sạn. Bằng cách sử dụng các thuật toán như Cây quyết định (Decision Tree) và Rừng ngẫu nhiên (Random Forest), chúng em đạt được độ chính xác rất cao, lên đến 100% trên cả tập huấn luyện và tập kiểm tra. Điều này cho thấy tiềm năng lớn của học máy trong việc hỗ trợ quản lý đặt phòng và tối ưu hóa hoạt động kinh doanh khách sạn.

Phân tích dữ liệu ban đầu giúp xác định các yếu tố quan trọng ảnh hưởng đến việc hủy đặt phòng, bao gồm thời gian đặt trước (lead_time), lịch sử hủy đặt phòng, số lượng yêu cầu đặc biệt và kênh đặt phòng. Những yếu tố này có thể được sử dụng để cải thiện chính sách đặt phòng cũng như thiết kế các biện pháp khuyến khích khách hàng duy trì đặt phòng của họ.

Tuy nhiên, một vấn đề quan trọng trong mô hình là nguy cơ quá khớp (overfitting). Độ chính xác tuyệt đối có thể là dấu hiệu cho thấy mô hình đã học quá kỹ từ dữ liệu huấn luyện, khiến nó khó thích nghi với dữ liệu mới. Để khắc phục vấn đề này, có thể sử dụng kỹ thuật pruning, điều chỉnh tham số mô hình hoặc thử nghiệm với các thuật toán khác như Gradient Boosting hoặc XGBoost.

Một hạn chế khác của nghiên cứu là khả năng giải thích của mô hình. Trong khi Cây quyết định dễ hiểu, thì Rừng ngẫu nhiên và các mô hình phức tạp hơn có thể khó giải thích đối với nhân viên khách sạn. Việc áp dụng các phương pháp như SHAP (SHapley Additive Explanations) hoặc LIME (Local Interpretable Model-agnostic Explanations) có thể giúp làm rõ ảnh hưởng của từng yếu tố lên kết quả dự báo.

Bên cạnh đó, dữ liệu đầu vào có thể chưa đầy đủ, khi chưa xem xét đến các yếu tố như đánh giá của khách hàng, yếu tố mùa vụ hay các sự kiện đặc biệt. Việc mở rộng tập dữ liệu và thu thập thêm thông tin có thể giúp cải thiện độ chính xác và tính ứng dụng của mô hình.

Trong thực tế, các mô hình dự báo hủy đặt phòng có thể hỗ trợ khách sạn trong việc tối ưu hóa chính sách giá, cải thiện quản lý nguồn lực và giảm thiểu tác động tiêu cực từ các

lượt hủy phòng. Hơn nữa, mô hình này có thể mở rộng sang các lĩnh vực khác như hàng không, đặt vé sự kiện hay dịch vụ thuê xe, nơi tình trạng hủy đặt chỗ cũng ảnh hưởng đến doanh thu.