**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**APACHE SPARK**

**Giảng viên hướng dẫn:**  Ngô Dương Hà

**Các thành viên trong nhóm:**

Nguyễn Vĩ Ân – 2001210529 – 12DHTH13

Nguyễn Trung Thành – 2001216049 – 12DHTH16

*Thứ năm ngày 24 tháng 09 năm 2024*

**MỤC LỤC**

[**Apache Spark trong Big Data** 3](#_Toc180684439)

[1. Lịch sử Apache Spark 3](#_Toc180684440)

[1.1 Bối cảnh: 3](#_Toc180684441)

[1.2 Phát triển: 3](#_Toc180684442)

[2. Giới thiệu đề tài 3](#_Toc180684443)

[2.1 Bối cảnh dự án: 3](#_Toc180684444)

[2.2 Mục tiêu của dự án: 4](#_Toc180684445)

[2.1 Tổng quan về công nghệ sử dụng: 4](#_Toc180684446)

[3. Quy trình thực hiện 4](#_Toc180684447)

[3.1 Thu thập và lưu trữ dữ liệu với MinIO 4](#_Toc180684448)

[3.2 Xử lý dữ liệu với Apache Spark: 5](#_Toc180684449)

[3.3 Xây dựng mô hình dự đoán: 5](#_Toc180684450)

[3.4 Triển khai ứng dụng với Streamlit: 6](#_Toc180684451)

[3.5 Lưu trữ và quản lý kết quả dự đoán với MinIO 6](#_Toc180684452)

[4. Kết quả và phân tích 7](#_Toc180684453)

[4.1 Đánh giá mô hình dự đoán: 7](#_Toc180684454)

[4.2 Phân tích các yếu tố ảnh hưởng đến khả năng sống sót 7](#_Toc180684455)

[5. Kết luận và định hướng phát triển 7](#_Toc180684456)

[**TÀI LIỆU THAM KHẢO** 9](#_Toc180684457)

# **Apache Spark trong Big Data**

Apache Spark là một nền tảng xử lý dữ liệu mạnh mẽ và nhanh chóng, thường được sử dụng trong lĩnh vực Big Data. Spark cung cấp một giao diện lập trình dễ sử dụng và được tối ưu hóa cho các công việc xử lý dữ liệu lớn, phân tích và tính toán phân tán trên các cụm máy chủ.

## Lịch sử Apache Spark

### Bối cảnh:

* ***Apache Spark*** ra đời vào năm 2009 tại UC Berkeley do Matei Zaharia khởi xướng, nhằm khắc phục hạn chế của Hadoop MapReduce. Spark nổi bật với khả năng xử lý dữ liệu nhanh nhờ lưu trữ trong bộ nhớ (in-memory), giúp tăng tốc độ xử lý các tác vụ như học máy, phân tích thời gian thực, và xử lý đồ thị.

### Phát triển:

* ***Năm 2013***, Spark gia nhập Apache Software Foundation và nhanh chóng trở thành một trong những công cụ Big Data phổ biến nhất. Công ty Databricks, do nhóm nghiên cứu thành lập, đã hỗ trợ mạnh mẽ cho sự phát triển của Spark, biến nó thành công cụ chủ chốt trong việc xử lý và phân tích dữ liệu lớn.

## Giới thiệu đề tài

*Tên đề tài*: “**DỰ ĐOÁN KHẢ NĂNG SỐNG SÓT TRONG THẢM HỌA TITANIC**

**SỬ DỤNG APACHE SPARK**”

### Bối cảnh dự án:

Thảm họa tàu Titanic vào năm 1912 đã trở thành một trong những sự kiện nổi tiếng nhất trong lịch sử nhân loại. Với hơn 1.500 người thiệt mạng, đây là một trong những vụ đắm tàu thương mại lớn nhất. Bộ dữ liệu về hành khách trên Titanic đã được sử dụng rộng rãi trong các dự án khoa học dữ liệu để dự đoán khả năng sống sót của hành khách dựa trên các yếu tố như tuổi, giới tính, và hạng vé. Trong dự án này, chúng tôi ứng dụng Apache Spark để xử lý dữ liệu lớn, sử dụng Streamlit để xây dựng giao diện người dùng, và MinIO để lưu trữ dữ liệu.

### Mục tiêu của dự án:

Dự án này nhằm xây dựng một mô hình dự đoán khả năng sống sót của hành khách trên tàu Titanic dựa trên các đặc điểm cá nhân như giới tính, độ tuổi, hạng vé, và các yếu tố khác. Sử dụng Apache Spark để xử lý dữ liệu, Streamlit để triển khai giao diện tương tác, và MinIO để lưu trữ dữ liệu. Mục tiêu là tạo ra một hệ thống dễ dàng mở rộng và tích hợp các công nghệ mạnh mẽ trong xử lý dữ liệu lớn.

### Tổng quan về công nghệ sử dụng:

* **Apache Spark:** Là một công cụ xử lý dữ liệu mạnh mẽ, đặc biệt hiệu quả trong việc xử lý dữ liệu lớn với khả năng tính toán song song. Nó cung cấp API cho nhiều ngôn ngữ, bao gồm Python, và tích hợp sẵn các thư viện học máy trong Spark MLlib.
* **Streamlit**: Là một framework mã nguồn mở giúp xây dựng giao diện người dùng trực quan và tương tác chỉ với một vài dòng mã Python. Công cụ này giúp triển khai nhanh các ứng dụng dựa trên học máy mà không cần kiến thức sâu về phát triển web.
* **MinIO**: Là một hệ thống lưu trữ đối tượng phân tán tương thích với Amazon S3. MinIO cung cấp một nền tảng linh hoạt và mạnh mẽ để lưu trữ dữ liệu lớn, đồng thời có thể tích hợp tốt với các ứng dụng phân tích dữ liệu như Spark.

## Quy trình thực hiện

### Thu thập và lưu trữ dữ liệu với MinIO

Bộ dữ liệu Titanic được lấy từ nguồn mở (Kaggle) và lưu trữ trên MinIO. Việc sử dụng MinIO giúp đảm bảo tính khả dụng và khả năng mở rộng khi làm việc với các tập dữ liệu lớn.

Lưu trữ: MinIO hoạt động như một hệ thống lưu trữ phân tán, cho phép truy cập dữ liệu từ nhiều nguồn khác nhau. Dữ liệu Titanic được lưu trữ dưới dạng các file CSV trong MinIO.

Tích hợp: Spark có thể tích hợp trực tiếp với MinIO thông qua các API tương thích S3, cho phép chúng ta tải dữ liệu từ MinIO vào môi trường Spark để xử lý.

### Xử lý dữ liệu với Apache Spark:

Dữ liệu Titanic chứa nhiều biến liên quan đến hành khách như tuổi, giới tính, hạng vé, và thông tin về gia đình đi kèm. Quá trình xử lý dữ liệu bao gồm:

* **Làm sạch dữ liệu**: Dữ liệu ban đầu chứa một số giá trị thiếu hoặc không hợp lệ, đặc biệt là trong cột Age. Các giá trị thiếu được xử lý bằng cách sử dụng phương pháp điền giá trị trung bình hoặc loại bỏ các dòng dữ liệu không hợp lệ.
* **Chuẩn hóa dữ liệu**: Một số cột như giới tính (Sex) và nơi khởi hành (Embarked) là các biến phân loại. Chúng được chuyển đổi thành các dạng số để mô hình học máy có thể sử dụng dễ dàng hơn.
* **Phân tích thống kê**: Dữ liệu được phân tích để hiểu rõ mối quan hệ giữa các thuộc tính và khả năng sống sót. Các biểu đồ và bảng phân phối cho thấy giới tính và hạng vé là những yếu tố quan trọng ảnh hưởng đến khả năng sống sót.

### Xây dựng mô hình dự đoán:

Sử dụng Apache Spark, xây dựng các mô hình học máy để dự đoán khả năng sống sót của hành khách dựa trên các thuộc tính:

* **Logistic Regression**: Đây là một thuật toán phân loại nhị phân, rất phù hợp để dự đoán khả năng sống sót (sống hoặc chết). Mô hình được huấn luyện trên tập dữ liệu đã được làm sạch và chuẩn hóa.
* **Decision Tree**: Một thuật toán phân loại khác được sử dụng để kiểm tra xem liệu nó có thể cải thiện độ chính xác dự đoán so với Logistic Regression.
* **Random Forest**: giúp cải thiện độ chính xác mà còn đối phó tốt hơn với sự phân bố không đều của các thuộc tính như giới tính, hạng vé, và độ tuổi của hành khách.
* **Đánh giá mô hình**: Mô hình Logistic Regression đạt độ chính xác khoảng 80%, trong khi Decision Tree có độ chính xác khoảng 75%. Mô hình Logistic Regression có hiệu quả hơn trong trường hợp dữ liệu Titanic do sự phân bố không đều của các yếu tố như giới tính và hạng vé. Mô hình *Random Forest* cho thấy độ chính xác khoảng 82%, cao hơn cả *Logistic Regression* và *Decision Tree*.

### Triển khai ứng dụng với Streamlit:

Sau khi xây dựng mô hình dự đoán, chúng tôi đã sử dụng Streamlit để triển khai một giao diện người dùng đơn giản và thân thiện. Các tính năng của ứng dụng bao gồm:

* **Giao diện người dùng**: Người dùng có thể nhập các thông tin cá nhân như giới tính, độ tuổi, và hạng vé. Dựa trên các thông tin này, ứng dụng sẽ sử dụng mô hình đã huấn luyện để dự đoán liệu hành khách có sống sót hay không.
* **Trực quan hóa kết quả**: Streamlit cho phép hiển thị trực quan các biểu đồ và kết quả phân tích, giúp người dùng dễ dàng hiểu được các yếu tố ảnh hưởng đến khả năng sống sót.

### Lưu trữ và quản lý kết quả dự đoán với MinIO

Kết quả dự đoán và các mô hình học máy sau khi huấn luyện được lưu trữ trong MinIO. Điều này đảm bảo rằng dữ liệu luôn có sẵn và có thể truy cập từ nhiều ứng dụng khác nhau, bao gồm cả những ứng dụng khác sử dụng Spark hoặc hệ thống phân tích dữ liệu lớn khác.

## Kết quả và phân tích

### Đánh giá mô hình dự đoán:

Mô hình Logistic Regression đã cho kết quả tốt hơn so với mô hình Decision Tree, với độ chính xác đạt khoảng 80%. Điều này cho thấy mô hình này hiệu quả trong việc phân loại khả năng sống sót dựa trên các yếu tố đầu vào.

### Phân tích các yếu tố ảnh hưởng đến khả năng sống sót

Sau khi phân tích mô hình cho thấy một số yếu tố ảnh hưởng mạnh mẽ đến khả năng sống sót của hành khách:

* **Giới tính**: Hành khách nữ có tỷ lệ sống sót cao hơn nam giới. Điều này có thể được giải thích bởi quy tắc "phụ nữ và trẻ em được cứu trước" trong tình huống khẩn cấp.
* **Hạng vé**: Hành khách ở hạng vé 1 (hạng cao cấp) có tỷ lệ sống sót cao hơn so với những hành khách ở hạng 3. Điều này cho thấy sự khác biệt về ưu tiên trong việc sơ tán.
* **Tuổi tác**: Trẻ em có tỷ lệ sống sót cao hơn người lớn, đặc biệt là những người lớn tuổi. Điều này cũng phù hợp với quy tắc ưu tiên cứu trẻ em.

### Kết luận và định hướng phát triển

Dự án này đã chứng minh tính hiệu quả của việc sử dụng Apache Spark để xử lý dữ liệu lớn và tích hợp với các công nghệ hiện đại như MinIO và Streamlit. Một hệ thống phân tích và dự đoán đã được xây dựng thành công, với khả năng dự đoán chính xác khả năng sống sót của hành khách Titanic dựa trên các yếu tố cá nhân.

Định hướng phát triển trong tương lai:

* **Tăng cường độ chính xác của mô hình**: Cải thiện mô hình bằng cách sử dụng thêm các thuật toán học máy phức tạp hơn như Random Forest, Gradient Boosting, hoặc Deep Learning.
* **Mở rộng tính năng của ứng dụng**: Bổ sung thêm nhiều tính năng tương tác, chẳng hạn như hiển thị các thống kê chi tiết hoặc cho phép người dùng tải lên tập dữ liệu riêng của họ để dự đoán.
* **Tích hợp các hệ thống lưu trữ khác**: Ngoài MinIO, có thể mở rộng việc lưu trữ và xử lý dữ liệu với các giải pháp đám mây khác như AWS S3 hoặc Google Cloud Storage.

# **TÀI LIỆU THAM KHẢO**

1. Tập titanic: <https://www.kaggle.com/c/titanic>
2. MinIO: <https://min.io/download?license=enterprise&platform=windows>
3. Apache Spark: <https://spark.apache.org/downloads.html>
4. Python: <https://www.python.org/downloads/>
5. Phân tích: <https://www.kaggle.com/code/nguyenvian0529/khaithacdulieu-logistic>
6. Mô hình khác: <https://www.kaggle.com/code/nguyenvian0529/dulieukhac>