**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH KHOA CÔNG NGHỆ THÔNG TIN**

****

**DỮ LIỆU LỚN**

**Tên đề tài: Sử dụng spark dự doán chất lượng sữa**

**Giảng viên hướng dẫn: Phạm Đình Tài Sinh viên thực hiện: Trần Đức Đại**

**MSSV: 2200003458**

**Khoá: 22**

**Ngành/ Chuyên ngành: Khoa học dữ liệu**

TP. HCM, tháng 9 năm 2025

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH KHOA CÔNG NGHỆ THÔNG TIN**

****

**DỮ LIỆU LỚN**

**Tên đề tài: Sử dụng spark dự doán chất lượng sữa**

**Giảng viên hướng dẫn: Phạm Đình Tài Sinh viên thực hiện: Trần Đức Đại**

**MSSV: 2200003458**

**Khoá: 22**

**Ngành/ Chuyên ngành: Khoa học dữ liệu**

TP. HCM, tháng 9 năm 2025

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến giảng viên Phạm Đình Tài, người đã trực tiếp giảng dạy và hướng dẫn em trong suốt quá trình học tập môn "Dữ Liệu Lớn" tại Đại học Nguyễn Tất Thành. Thầy đã cung cấp cho tôi những kiến thức sâu rộng về dữ liệu, đặc biệt là trong các ứng dụng thực tế của dữ liệu. Sự nhiệt tình, tận tâm và am hiểu của Thầy đã giúp em có cái nhìn rõ ràng hơn về các phương pháp, kỹ thuật và mô hình học sâu, đồng thời khơi dậy niềm đam mê nghiên cứu và học hỏi trong em.

Mỗi bài giảng của Thầy đều mang lại những kiến thức quý báu, không chỉ giúp em hiểu rõ hơn về các lý thuyết nền tảng mà còn hướng dẫn tôi áp dụng những kiến thức đó vào thực tiễn. Thầy cũng luôn sẵn sàng giải đáp những thắc mắc và cung cấp những lời khuyên quý giá trong quá trình thực hiện dự án của em. Những giờ học, những bài giảng của Thầy không chỉ giúp em cải thiện kỹ năng học thuật mà còn trang bị cho em những kỹ năng nghiên cứu, phân tích và giải quyết vấn đề trong lĩnh dữ liệu.

Cuối cùng, em xin chân thành cảm ơn Thầy Phạm Đình Tài vì những đóng góp to lớn trong việc hình thành và phát triển kiến thức của em. Hy vọng rằng tôi sẽ tiếp tục áp dụng những kiến thức mà Thầy truyền đạt để đóng góp vào sự phát triển của ngành khoa học dữ liệu trong tương lai.

Trân trọng cảm ơn!

# LỜI MỞ ĐẦU

Chất lượng sữa là một yếu tố then chốt, ảnh hưởng trực tiếp đến sức khỏe người tiêu dùng và hiệu quả kinh doanh của ngành công nghiệp chế biến sữa. Tuy nhiên, việc đánh giá chất lượng sữa một cách nhanh chóng, chính xác và hiệu quả vẫn luôn là một thách thức. Các phương pháp kiểm tra truyền thống thường tốn kém và mất nhiều thời gian, gây ảnh hưởng đến quy trình sản xuất và phân phối.

Để giải quyết vấn đề này, dự án này tập trung vào việc ứng dụng công nghệ **Spark** để xây dựng một hệ thống dự đoán chất lượng sữa. Cụ thể, chúng tôi sử dụng các mô hình học máy để phân tích các yếu tố hóa lý của sữa, từ đó đưa ra dự đoán về chất lượng. Bằng cách tận dụng sức mạnh của Spark, chúng tôi có thể xử lý các tập dữ liệu lớn một cách hiệu quả, giúp quá trình dự đoán diễn ra nhanh chóng và đáng tin cậy.

Mục tiêu chính của dự án là xây dựng và đánh giá các mô hình dự đoán, bao gồm **Logistic Regression**, **Decision Tree** và **Random Forest**, nhằm tìm ra mô hình tối ưu nhất. Kết quả của dự án sẽ không chỉ cung cấp một công cụ hữu ích cho ngành công nghiệp sữa mà còn đóng góp vào việc ứng dụng khoa học dữ liệu và học máy trong lĩnh vực nông nghiệp và công nghiệp thực phẩm.

BM-ChT-

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN HỌC KỲ III NĂM HỌC 2024 – 2025** |

## PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN

Môn thi: Dữ Liệu Lớn Lớp học phần: 22DTH4B

Sinh viên thực hiện : Trần Đức Đại Đóng góp:100%

Ngày thi: 05/09/2025 Phòng thi: L.

Đề tài tiểu luận/báo cáo của sinh viên :

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí (theo**  **CĐR HP)** | **Đánh giá của GV** | **Điểm tối**  **đa** | **Điểm đạt**  **được** |
| Cấu trúc của báo cáo |  | 1 |  |
| **Nội dung** |  |  |  |
| - Các nội dung thành phần |  | 6 |  |
| - Lập luận |  | 2 |  |
| - Kết luận |  | 0.5 |  |
| Trình bày |  | 0.5 |  |
| **TỔNG ĐIỂM** |  | **10** |  |

# MỤC LỤC

[LỜI CẢM ƠN 1](#_bookmark0)

[LỜI MỞ ĐẦU 2](#_bookmark1)

[MỤC LỤC 4](#_bookmark2)

[DANH MỤC HÌNH 6](#_bookmark3)

[KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT 7](#_bookmark4)

[CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI 8](#_bookmark5)

1. [Giới thiệu đề tài 8](#_bookmark6)
   1. [Khái niệm 8](#_bookmark7)
      1. [Khái niệm về Chất lượng sữa 8](#_bookmark8)
      2. [Các chỉ số đánh giá chất lượng sữa 8](#_bookmark9)
      3. [Vấn đề về dữ liệu trong ngành sữa 8](#_bookmark10)
   2. [Giới thiệu đề tài 9](#_bookmark11)
2. [Lý do chọn đề tài 9](#_bookmark12)
3. [Mục tiêu của đề tài 10](#_bookmark14)
4. [Các Hướng Tiếp Cận và Giải Quyết Bài Toán 12](#_bookmark15)
   1. [Thu thập và Tiền xử lý Dữ liệu 12](#_bookmark16)
   2. [Xây dựng và Huấn luyện Mô hình 13](#_bookmark17)
   3. [Đánh giá và So sánh Kết quả 13](#_bookmark18)
   4. [Đưa ra Giải pháp và Ứng dụng Thực tiễn 14](#_bookmark19)
   5. [Phương pháp Machine Learning truyền thống 15](#_bookmark20)
   6. [Trích chọn đặc trưng 16](#_bookmark22)
5. [Làm quen với PySpark 16](#_bookmark23)
   1. [Models 17](#_bookmark25)
      1. [Linear Models (Mô hình tuyến tính): 18](#_bookmark27)
      2. [Decision Trees (Cây quyết định): 18](#_bookmark28)
      3. [Ensemble Models (Mô hình tổ hợp): 18](#_bookmark29)
      4. [Clustering (Phân cụm): 18](#_bookmark30)
      5. [Collaborative Filtering (Lọc cộng tác) 18](#_bookmark31)
   2. [Tiền xử lý 18](#_bookmark32)
   3. [Note 19](#_bookmark33)
6. [Pyspark trong Python là gì? 19](#_bookmark34)
   1. [Lịch sử ra đời Pyspark 20](#_bookmark36)
   2. [Kiến trúc của Pyspark 21](#_bookmark37)
   3. [Cách Pyspark hoạt động 21](#_bookmark38)
   4. [Lợi ích từ Pyspark 22](#_bookmark39)
7. [Đối tượng và phạm vi nghiên cứu 22](#_bookmark40)
   1. [Mục đích nghiên cứu 22](#_bookmark41)
   2. [Đối tượng và phạm vi nghiên cứu 22](#_bookmark42)

[CHƯƠNG 2: ỨNG DỤNG THUẬT TOÁN 23](#_bookmark43)

1. [Mô tả bài toán 23](#_bookmark44)
2. [Xây dựng bộ dữ liệu 25](#_bookmark46)
3. [Áp dụng thuật toán 26](#_bookmark47)
   1. [Giới thiệu về Hồi quy Logistic 27](#_bookmark48)
   2. [Cơ chế Hoạt động trong Bài toán Dự đoán Chất lượng Sữa 27](#_bookmark49)
   3. [Lợi ích của việc sử dụng Hồi quy Logistic 27](#_bookmark50)

[CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG 29](#_bookmark51)

* + 1. [Khởi tạo Môi trường Spark 29](#_bookmark52)
    2. [Tiền xử lý Dữ liệu 30](#_bookmark54)
    3. [Huấn luyện và Đánh giá Mô hình 31](#_bookmark55)

[CHƯƠNG 4. KẾT LUẬN 33](#_bookmark56)

1. [Kết quả đạt được 33](#_bookmark57)
2. [Hạn chế và hướng phát triển 34](#_bookmark58)
   1. [Hạn chế 34](#_bookmark59)
   2. [Hướng phát triển 34](#_bookmark60)

[KẾT LUẬN 36](#_bookmark61)

[DANH MỤC TÀI LIỆU THAM KHẢO 37](#_bookmark62)

# DANH MỤC HÌNH

[Hình 1. Minh họa bài toán 10](#_bookmark13)

[Hình 2: Training Phase 15](#_bookmark21)

[Hình 3: PySpark 17](#_bookmark24)

[Hình 4: Thực thi qua Logistic Regression (Hồi quy) 17](#_bookmark26)

[Hình 5: Uư điểm của PySpark trong Python 20](#_bookmark35)

[Hình 6. Minh họa sữa 23](#_bookmark45)

[Hình 7. Data PIPELINE 29](#_bookmark53)

# KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

|  |  |
| --- | --- |
| Chữ viết tắt | Ý nghĩa |
| ITS | Intelligent Transportation System (Hệ thống Giao thông Thông minh) |
| GPS | Global Positioning System (Hệ thống Xác định Vị trí Toàn cầu) |
| ARIMA | Autoregressive Integrated Moving Average (Mô hình Tự hồi quy Kết hợp Trung bình di chuyển) |
| API | Application Programming Interface (Giao diện Lập trình Ứng dụng) |
| LSTM | Long Short-Term Memory (Bộ Nhớ Dài Ngắn hạn) |

# CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu đề tài

### Khái niệm

### Khái niệm về Chất lượng sữa

Chất lượng sữa là tổng hợp các yếu tố và thuộc tính của sữa, được đánh giá dựa trên các tiêu chuẩn vật lý, hóa học và vi sinh vật. Các chỉ số này quyết định độ tươi, an toàn và giá trị dinh dưỡng của sữa. Việc đánh giá chất lượng sữa một cách chính xác và kịp thời là vô cùng quan trọng đối với toàn bộ chuỗi cung ứng, từ trang trại đến người tiêu dùng. Nó không chỉ đảm bảo sức khỏe cộng đồng mà còn tối ưu hóa hiệu quả sản xuất và kinh doanh.

### Các chỉ số đánh giá chất lượng sữa

Chất lượng sữa được xác định dựa trên một số chỉ số chính, bao gồm:

* + - * Độ pH: Chỉ số axit hoặc kiềm trong sữa. pH của sữa tươi thường dao động quanh 6.7.
      * Nhiệt độ: Ảnh hưởng đến sự phát triển của vi khuẩn và quá trình bảo quản. Nhiệt độ thấp giúp duy trì độ tươi của sữa.
      * Độ béo (Fat): Hàm lượng chất béo trong sữa, một trong những chỉ số quan trọng nhất về giá trị dinh dưỡng.
      * Độ ẩm (Moisture): Tỷ lệ nước trong sữa, thường được kiểm tra để đảm bảo sữa không bị pha loãng.
      * Mật độ (Density): Tỷ trọng của sữa, dùng để kiểm tra độ nguyên chất.
      * Hàm lượng muối (Salt): Chỉ số này có thể phản ánh tình trạng sức khỏe của con vật và ảnh hưởng đến hương vị của sữa.
      * Hàm lượng đạm (Protein): Một chỉ số dinh dưỡng quan trọng, cần đạt một mức nhất định.
      * Hàm lượng lactose: Đường sữa tự nhiên, là nguồn năng lượng chính.

### Vấn đề về dữ liệu trong ngành sữa

Trong sản xuất sữa hiện đại, lượng dữ liệu về chất lượng sữa được thu thập rất lớn và liên tục từ các cảm biến, thiết bị kiểm tra tự động. Các dữ liệu này thường có khối lượng lớn (Big Data), đòi hỏi các công cụ phân tích mạnh mẽ và hiệu quả để xử lý, chẳng hạn như Apache Spark.

### Giới thiệu đề tài

Nhu cầu về thực phẩm sạch và an toàn ngày càng tăng cao, đặc biệt là các sản phẩm sữa. Do đó, việc kiểm soát và dự đoán chất lượng sữa trở thành một yêu cầu cấp thiết. Các phương pháp kiểm tra truyền thống thường tốn kém và mất nhiều thời gian, có thể làm gián đoạn quy trình sản xuất và vận chuyển. Để giải quyết vấn đề này, việc ứng dụng công nghệ Big Data và học máy để phân tích và dự đoán chất lượng sữa là một giải pháp đột phá.

Đề tài này tập trung vào việc xây dựng một hệ thống dự đoán chất lượng sữa sử dụng các thuật toán học máy trên nền tảng Apache Spark. Chúng tôi sẽ phân tích các đặc trưng của sữa như độ pH, nhiệt độ, độ béo, v.v., để dự đoán chất lượng sữa (tươi, đã hỏng, v.v.). Mục tiêu là xây dựng một mô hình có độ chính xác cao, có khả năng xử lý dữ liệu lớn, giúp các doanh nghiệp trong ngành sữa có thể đưa ra các quyết định nhanh chóng và hiệu quả, giảm thiểu rủi ro và tối ưu hóa quy trình sản xuất.

## Lý do chọn đề tài

Việc đảm bảo chất lượng sản phẩm là một trong những ưu tiên hàng đầu của ngành công nghiệp thực phẩm, và ngành sữa cũng không ngoại lệ. Tuy nhiên, việc kiểm soát chất lượng sữa đối mặt với nhiều thách thức lớn. Các phương pháp kiểm tra truyền thống, dựa trên phân tích hóa học và vi sinh vật học trong phòng thí nghiệm, tuy chính xác nhưng lại tốn kém về chi phí, thời gian và nguồn nhân lực. Điều này đặc biệt bất lợi đối với các doanh nghiệp sản xuất quy mô lớn, nơi hàng triệu lít sữa được xử lý mỗi ngày. Một lô sữa kém chất lượng, nếu không được phát hiện kịp thời, có thể gây ra những thiệt hại kinh tế nghiêm trọng và ảnh hưởng tiêu cực đến sức khỏe người tiêu dùng, làm suy giảm uy tín thương hiệu.

Nhận thức được những hạn chế đó, việc tìm kiếm một giải pháp hiệu quả hơn trở nên cấp thiết. Sự ra đời của các công nghệ Big Data và học máy đã mở ra một hướng đi mới đầy tiềm năng. Ngày nay, các trang trại chăn nuôi và nhà máy chế biến sữa đều trang bị hệ thống cảm biến thông minh, thu thập hàng loạt dữ liệu về chất lượng sữa theo thời gian thực: từ nhiệt độ, độ pH, đến mật độ, hàm lượng chất béo và protein. Lượng dữ liệu này tăng lên theo cấp số nhân, tạo thành một kho dữ liệu khổng lồ mà các công cụ phân tích truyền thống không thể xử lý. Đây chính là lúc Apache Spark phát huy sức mạnh. Spark, với khả năng xử lý song song và tính toán phân tán, cho phép chúng ta xử lý và phân tích hàng petabyte dữ liệu một cách nhanh chóng, hiệu quả và đáng tin cậy.

Việc kết hợp Spark với các thuật toán học máy giúp chúng ta không chỉ phân tích mà còn dự đoán chất lượng sữa trước khi nó đến tay người tiêu dùng.



*Hình 1. Minh họa bài toán*

Hơn nữa, việc áp dụng học máy vào bài toán này mang lại những lợi ích vượt trội. Các mô hình học máy có khả năng phát hiện các mối quan hệ phức tạp và những quy luật ẩn trong dữ liệu mà con người khó có thể nhận ra. Bằng cách huấn luyện mô hình trên dữ liệu lịch sử, chúng ta có thể xây dựng một hệ thống tự động dự đoán chất lượng sữa dựa trên các chỉ số đầu vào, giúp các doanh nghiệp chủ động hơn trong việc kiểm soát chất lượng. Hệ thống này không chỉ giúp tiết kiệm chi phí và thời gian mà còn giảm thiểu rủi ro, bảo vệ người tiêu dùng, và nâng cao vị thế cạnh tranh của doanh nghiệp trên thị trường.

## Mục tiêu của đề tài

Mục tiêu chính của đề tài này là xây dựng một hệ thống dự đoán chất lượng sữa tự động, hiệu quả và có khả năng mở rộng. Để đạt được mục tiêu đó, chúng tôi đặt ra các mục tiêu cụ thể như sau:

Nghiên cứu và Phân tích Dữ liệu: Đây là bước nền tảng, đảm bảo tính chính xác cho toàn bộ dự án. Chúng tôi sẽ tiến hành nghiên cứu sâu rộng về bộ dữ liệu, bao gồm các thuộc tính vật lý và hóa học của sữa như pH, nhiệt độ, độ béo, độ ẩm, mật độ, và các chỉ số khác. Sau khi thu thập dữ liệu, chúng tôi sẽ thực hiện các bước tiền xử lý chuyên sâu, bao gồm việc làm sạch dữ liệu để loại bỏ các giá trị nhiễu hoặc không hợp lệ, xử lý

các giá trị bị thiếu bằng các phương pháp thống kê như điền giá trị trung bình hoặc median, và chuẩn hóa dữ liệu để các thuộc tính có cùng một thang đo, giúp các thuật toán học máy hoạt động hiệu quả hơn.

Xây dựng và Lựa chọn Mô hình Học máy: Sau khi dữ liệu được chuẩn bị sẵn sàng, chúng tôi sẽ tiến hành xây dựng các mô hình dự đoán. Đề tài sẽ tập trung vào việc áp dụng và so sánh hiệu suất của ba thuật toán học máy phổ biến và mạnh mẽ trên nền tảng Spark MLlib:

Logistic Regression: Một thuật toán phân loại tuyến tính đơn giản nhưng hiệu quả, thường được sử dụng làm mô hình cơ sở.

Decision Tree: Một mô hình trực quan, dễ hiểu và mạnh mẽ, có khả năng xử lý cả dữ liệu số và dữ liệu phân loại.

Random Forest: Một thuật toán mạnh hơn, kết hợp nhiều cây quyết định để đưa ra dự đoán chính xác hơn và giảm thiểu hiện tượng quá khớp (overfitting).

Đánh giá Hiệu suất và So sánh Mô hình: Để đảm bảo tính khách quan, chúng tôi sẽ chia dữ liệu thành tập huấn luyện và tập kiểm tra. Sau đó, các mô hình sẽ được đánh giá kỹ lưỡng bằng các chỉ số hiệu suất tiêu chuẩn trong lĩnh vực học máy. Các chỉ số này bao gồm:

Độ chính xác (Accuracy): Tỷ lệ các dự đoán đúng trên tổng số dự đoán.

Độ chính xác trên lớp tích cực (Precision): Trong số các mẫu được dự đoán là sữa tốt, có bao nhiêu mẫu thực sự là sữa tốt.

Độ thu hồi (Recall): Trong số các mẫu sữa tốt thực sự, mô hình đã dự đoán đúng bao nhiêu mẫu.

Độ đo F1 (F1-score): Một chỉ số kết hợp giữa Precision và Recall, rất hữu ích khi dữ liệu không cân bằng. Bằng cách so sánh các chỉ số này, chúng tôi sẽ tìm ra mô hình có hiệu suất tốt nhất, phù hợp nhất với bài toán.

Đóng góp và Ứng dụng Thực tiễn: Mục tiêu cuối cùng là mang lại giá trị thực tiễn cho ngành công nghiệp. Kết quả của dự án sẽ không chỉ là một mô hình lý thuyết mà còn là một giải pháp hữu ích. Chúng tôi sẽ đề xuất một quy trình và kiến trúc hệ thống để các doanh nghiệp có thể tích hợp mô hình này vào quy trình sản xuất và kiểm soát chất lượng của họ. Hệ thống này có thể hoạt động song song với các thiết bị cảm biến, tự động đưa ra cảnh báo khi phát hiện lô sữa có nguy cơ kém chất lượng, giúp các

doanh nghiệp có thể đưa ra quyết định nhanh chóng, giảm thiểu lãng phí và đảm bảo an toàn sản phẩm.

## Các Hướng Tiếp Cận và Giải Quyết Bài Toán

Để xây dựng một hệ thống dự đoán chất lượng sữa hiệu quả, chúng tôi sẽ thực hiện một quy trình gồm nhiều bước, từ thu thập dữ liệu thô đến triển khai mô hình cuối cùng. Quy trình này được thiết kế để đảm bảo tính chính xác và khả năng mở rộng của hệ thống, tận dụng tối đa sức mạnh của nền tảng Apache Spark.

### Thu thập và Tiền xử lý Dữ liệu

Đây là bước nền tảng, quyết định chất lượng của toàn bộ dự án. Dữ liệu thô về chất lượng sữa thường không hoàn hảo, có thể chứa nhiễu, các giá trị bị thiếu hoặc không đồng nhất. Chúng tôi sẽ tiếp cận vấn đề này một cách có hệ thống:

* Thu thập Dữ liệu: Nguồn dữ liệu chính cho dự án này là tập dữ liệu đã được thu thập từ các nguồn đáng tin cậy. Tập dữ liệu này bao gồm các thông số hóa lý quan trọng của sữa như: pH, Nhiệt độ, Độ béo (Fat), Độ ẩm (Moisture), Mật độ (Density), Hàm lượng muối (Salt), Hàm lượng đạm (Protein) và Hàm lượng lactose. Các thông số này đóng vai trò là các biến đầu vào để dự đoán chất lượng sữa đầu ra (tốt hoặc hỏng).
* Tiền xử lý Dữ liệu với Spark: Sau khi có dữ liệu, chúng tôi sẽ sử dụng Spark DataFrames để thực hiện các bước tiền xử lý một cách hiệu quả trên quy mô lớn.
* Loại bỏ Dữ liệu Nhiễu và Trùng lặp: Các bản ghi có giá trị bất thường hoặc trùng lặp sẽ được xác định và loại bỏ để đảm bảo tính toàn vẹn của dữ liệu.
* Xử lý Giá trị Thiếu: Một trong những thách thức phổ biến là các giá trị bị thiếu. Chúng tôi sẽ sử dụng các phương pháp như điền giá trị trung bình (mean) hoặc trung vị (median) cho các cột số, tùy thuộc vào phân bố dữ liệu.
* Chuẩn hóa Dữ liệu (Standardization/Normalization): Để các thuật toán học máy hoạt động tốt nhất, tất cả các đặc trưng (features) phải nằm trên cùng một thang đo. Chúng tôi sẽ sử dụng các kỹ thuật như StandardScaler hoặc MinMaxScaler của Spark ML để chuẩn hóa dữ liệu, đảm bảo rằng không có thuộc tính nào chiếm ưu thế không cần thiết so với các thuộc tính khác.
* VectorAssembler: Cuối cùng, các cột đặc trưng riêng lẻ sẽ được gộp lại thành một cột vector duy nhất bằng cách sử dụng VectorAssembler, một bước bắt buộc để các thuật toán học máy trong Spark MLlib có thể sử dụng dữ liệu.

### Xây dựng và Huấn luyện Mô hình

Với dữ liệu đã được chuẩn bị, chúng tôi sẽ tiến hành xây dựng và huấn luyện các mô hình học máy. Mục tiêu là tìm ra mô hình có khả năng dự đoán chất lượng sữa với độ chính xác cao nhất.

* Phân chia Dữ liệu: Dữ liệu sẽ được chia thành hai tập: tập huấn luyện (training set) (ví dụ: 80%) và tập kiểm tra (testing set) (20%). Tập huấn luyện được dùng để "dạy" mô hình, trong khi tập kiểm tra được giữ lại để đánh giá khách quan hiệu suất của mô hình trên dữ liệu mà nó chưa từng thấy.
* Lựa chọn Thuật toán: Chúng tôi sẽ thử nghiệm với ba mô hình phân loại phổ biến và hiệu quả, đều có sẵn trong thư viện Spark MLlib:
* Logistic Regression: Một thuật toán phân loại tuyến tính đơn giản nhưng mạnh mẽ, phù hợp cho các bài toán phân loại nhị phân như dự đoán chất lượng sữa (tốt/hỏng).
* Decision Tree: Thuật toán này tạo ra một "cây" các quyết định, giúp trực quan hóa cách mô hình đưa ra dự đoán. Đây là một mô hình rất hữu ích để hiểu được các yếu tố nào ảnh hưởng mạnh nhất đến chất lượng sữa.
* Random Forest: Đây là một thuật toán kết hợp (ensemble) mạnh mẽ, xây dựng nhiều cây quyết định độc lập và sau đó sử dụng phiếu bầu để đưa ra dự đoán cuối cùng. Random Forest thường cho kết quả chính xác cao hơn Decision Tree và giúp giảm thiểu hiện tượng quá khớp.
* Huấn luyện và Tinh chỉnh Mô hình: Các mô hình sẽ được huấn luyện trên tập huấn luyện. Chúng tôi cũng sẽ sử dụng các kỹ thuật tinh chỉnh tham số (hyperparameter tuning) như Grid Search để tìm ra các tham số tối ưu cho từng mô hình, ví dụ như độ sâu tối đa của cây quyết định hay số lượng cây trong Random Forest.

### Đánh giá và So sánh Kết quả

Để xác định mô hình tốt nhất, chúng tôi sẽ đánh giá hiệu suất của từng mô hình trên tập kiểm tra.

* Ma trận Nhầm lẫn (Confusion Matrix): Chúng tôi sẽ tạo ma trận nhầm lẫn để có cái nhìn tổng quan về hiệu suất của mô hình, bao gồm số lượng dự đoán đúng và sai cho cả hai lớp (sữa tốt và sữa hỏng).
* Các Chỉ số Hiệu suất Chính:
* Độ chính xác (Accuracy): Tỷ lệ tổng số dự đoán đúng.
* Precision: Tỷ lệ dự đoán đúng trong số các dự đoán là sữa "tốt". Chỉ số này quan trọng để tránh phân loại sai sữa hỏng thành sữa tốt.
* Recall: Tỷ lệ dự đoán đúng trong số các mẫu sữa "hỏng" thực sự. Chỉ số này quan trọng để đảm bảo tất cả các lô sữa hỏng đều được phát hiện.
* F1-score: Chỉ số tổng hợp giữa Precision và Recall, rất hữu ích khi tập dữ liệu không cân bằng.
* So sánh và Lựa chọn: Dựa trên các chỉ số trên, chúng tôi sẽ so sánh hiệu suất của Logistic Regression, Decision Tree và Random Forest để xác định mô hình nào hoạt động tốt nhất cho bài toán này. Kết quả này sẽ là cơ sở để chọn mô hình tối ưu nhất cho việc triển khai thực tiễn.

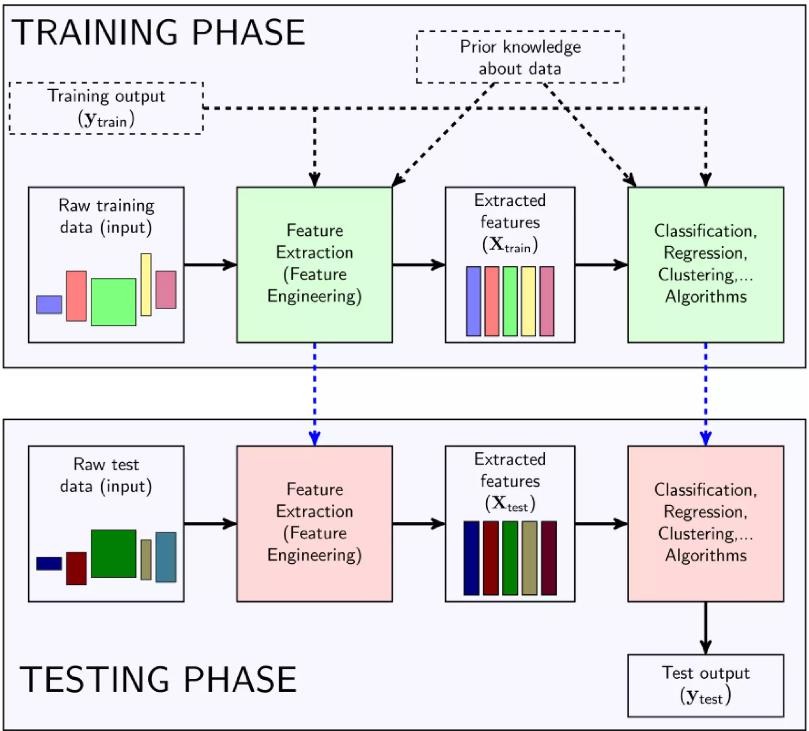
### Đưa ra Giải pháp và Ứng dụng Thực tiễn

Mục tiêu cuối cùng là biến kết quả nghiên cứu thành một giải pháp hữu ích.

* Xây dựng Quy trình Tự động: Chúng tôi sẽ xây dựng một quy trình tự động hóa việc dự đoán chất lượng sữa. Dữ liệu mới từ các cảm biến sẽ được đưa vào mô hình đã huấn luyện, và mô hình sẽ nhanh chóng đưa ra dự đoán về chất lượng.
* Hệ thống Cảnh báo Sớm: Kết quả dự đoán có thể được tích hợp vào một hệ thống cảnh báo sớm. Nếu mô hình dự đoán một lô sữa có khả năng hỏng, hệ thống sẽ tự động gửi cảnh báo đến bộ phận kiểm soát chất lượng, cho phép họ thực hiện kiểm tra sâu hơn hoặc đưa ra quyết định xử lý kịp thời, tránh phân phối sản phẩm kém chất lượng ra thị trường.
* Cải thiện Hiệu suất Sản xuất: Dữ liệu dự đoán sẽ giúp các nhà sản xuất hiểu rõ hơn về các yếu tố ảnh hưởng đến chất lượng sữa, từ đó đưa ra các điều chỉnh trong quy trình sản xuất, bảo quản và vận chuyển để nâng cao chất lượng sản phẩm một cách tổng thể.

### Phương pháp Machine Learning truyền thống

Mô hình hoạt động chung của các phương pháp Machine Learning được thể hiện trong hình dưới đây:



*Hình 2: Training Phase*

Ta có thể thấy Machine Learning gồm hai giai đoạn chính là Huấn luyện – Training và Thử nghiệm – Testing, trong mỗi giai đoạn đều sử dụng hai thành phần quan trọng nất do người xử lý bài toán thiết kế, đó là Trích chọn đặc trưng – Feature Engineering (hay còn gọi là Feature Extraction) và thuật toán phân loại, nhận dạng... – Algorithms. Hai thành phần này có ảnh hưởng trực tiếp đến kết quả bài toán, vì thế được thiết kế cẩn thận, tốn nhiều thời gian, đòi hỏi người thiết kế phải có kiến thức chuyên môn và nắm rõ đặc điểm của bài toán cần xử lý

### Trích chọn đặc trưng

Trong bài toán thực tế, ta chỉ có được những dữ liệu thô chưa qua chọn lọc xử lý, và để có thể đưa các dữ liệu này vào huấn luyện ta cần có những phép biến đổi để biến các dữ liệu thô thành dữ liệu chuẩn, không còn nhiều và có khả năng biểu diễn dữ liệu tốt hơn. Các thông tin đặc trưng này là khác nhau với từng loại dữ liệu và bài toán cụ thể, vì thế trong từng trường hợp phép biến đổi này cần phải được tùy biến một cách thích hợp để cải thiện độ chính xác của mô hình dự đoán. Quá trình này được gọi là Trích chọn đặc trưng

## Làm quen với PySpark

PySpark là một giao diện Python cho Apache Spark, một hệ thống xử lý dữ liệu phân tán và tính toán trên dữ liệu lớn. Apache Spark là một dự án mã nguồn mở phát triển nhanh chóng và trở thành một trong những Công cụ phân tích dữ liệu và tính toán phân tán phổ biến nhất. PySpark cho phép bạn sử dụng Python để tận dụng sức mạnh và tính phân tán của Spark trong việc xử lý và phân tích dữ liệu lớn. PySpark là một framework mạnh mẽ cho xử lý dữ liệu lớn và tích hợp sức mạnh của Apache Spark với ngôn ngữ lập trình Python. Dưới đây là một số ưu điểm của PySpark:

Xử lý dữ liệu lớn: PySpark được xây dựng trên Apache Spark, một framework phân tán được thiết kế để xử lý dữ liệu lớn. Nó cho phép bạn xử lý và phân tích dữ liệu có kích thước lớn một cách hiệu quả trên các cluster máy tính.

Tích hợp với Python: PySpark được tích hợp chặt chẽ với Python, ngôn ngữ lập trình phổ biến và dễ sử dụng. Điều này giúp người phân tích dữ liệu và nhà phát triển sử dụng Python có thể dễ dàng tận dụng sức mạnh của Apache Spark.

API Dễ Sử Dụng: PySpark cung cấp một API Python dễ sử dụng cho nhiều tác vụ, bao gồm xử lý dữ liệu, đào tạo mô hình học máy, và truy vấn dữ liệu. API này làm cho việc viết mã và phân tích dữ liệu trở nên thuận tiện.

Hỗ Trợ Đa Ngôn Ngữ: Ngoài Python, Spark hỗ trợ nhiều ngôn ngữ khác như Scala, Java và R. Điều này cho phép các nhóm làm việc với ngôn ngữ ưa thích của họ trong cùng một dự án.

Xử Lý Thời Gian Thực: Spark có khả năng xử lý dữ liệu thời gian thực, cho phép bạn xử lý và phản hồi nhanh chóng đối với dữ liệu đang chảy vào hệ thống.

Tối ưu hóa tự động: PySpark cung cấp trình tối ưu hóa tự động để cải thiện hiệu suất xử lý dữ liệu trên các cluster, giúp tận dụng tốt sức mạnh của các máy tính trong cluster.

Hệ sinh thái mạnh mẽ: Spark có một hệ sinh thái các thư viện và Công cụ bổ sung cho việc xử lý dữ liệu, machine learning, đồ họa, và nhiều tác vụ khác, tạo ra một môi trường phong phú cho phân tích dữ liệu và ứng dụng AI.

Cộng đồng và tài liệu phong phú: Spark và PySpark có một cộng đồng lớn, nhiều tài liệu học tập và hỗ trợ trực tuyến, điều này giúp bạn dễ dàng tìm kiếm giải pháp cho các thách thức cụ thể.

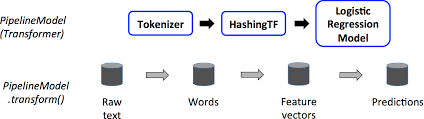


*Hình 3: PySpark*

Tóm lại, PySpark là một Công cụ mạnh mẽ và linh hoạt cho việc xử lý dữ liệu lớn và phân tích dữ liệu, đặc biệt là khi bạn muốn sử dụng Python và tirận hóa việc làm việc với Apache Spark.

### Models

PySpark là một giao diện Python cho Apache Spark, một hệ thống xử lý dữ liệu phân tán và tính toán trên dữ liệu lớn. Apache Spark là một dự án mã nguồn mở phát triển nhanh chóng và trở thành một trong những Công cụ phân tích dữ liệu và tính toán phân tán phổ biến nhất. PySpark cho phép bạn sử dụng Python để tận dụng sức mạnh và tính phân tán của Spark trong việc xử lý và phân tích dữ liệu lớn. PySpark là một framework mạnh mẽ cho xử lý dữ liệu lớn và tích hợp sức mạnh của Apache Spark với ngôn ngữ lập trình Python.



*Hình 4: Thực thi qua Logistic Regression (Hồi quy)*

### Linear Models (Mô hình tuyến tính):

Linear Regression (Hồi quy tuyến tính): Sử dụng để dự đoán giá trị số liệu liên tục dựa trên mối quan hệ tuyến tính giữa các biến đầu vào và biến mục tiêu.

Logistic Regression (Hồi quy logistic): Sử dụng để phân loại dữ liệu thành hai lớp (binary classification) dựa trên mối quan hệ logistic giữa biến đầu vào và biến phân loại.

### Decision Trees (Cây quyết định):

DecisionTreeClassifier: Sử dụng để phân loại dữ liệu thành các lớp dựa trên quyết định dựa trên các luật tại các nút của cây.

DecisionTreeRegressor: Sử dụng để dự đoán giá trị số liệu liên tục dựa trên cây quyết định.

### Ensemble Models (Mô hình tổ hợp):

Random Forest: Một tập hợp các cây quyết định để tăng tính chính xác và giảm overfitting.

Gradient-Boosted Trees (GBT): Tạo ra một tập hợp các cây quyết định theo cách tối ưu để tối đa hóa tính chính xác.

### Clustering (Phân cụm):

K-Means: Sử dụng để phân cụm dữ liệu thành các nhóm dựa trên sự tương tự giữa các điểm dữ liệu.

### Collaborative Filtering (Lọc cộng tác):

ALS (Alternating Least Squares): Sử dụng trong các hệ thống đề xuất để dự đoán sở thích của người dùng dựa trên lịch sử hoặc hành vi của họ và sử dụng sự cộng tác giữa người dùng.

### Tiền xử lý

Để tiền xử lý dữ liệu và huấn luyện mô hình trong PySpark, bạn cần thực hiện các bước sau:

* Đọc và tạo DataFrame chứa dữ liệu của bạn từ nguồn dữ liệu.
* Tiền xử lý dữ liệu bằng cách thực hiện các bước như lọc, chuyển đổi, chuẩn hóa dữ liệu.
* Chia dữ liệu thành bộ huấn luyện và bộ kiểm tra sử dụng phương thức

.randomSplit() hoặc .trainTestSplit().

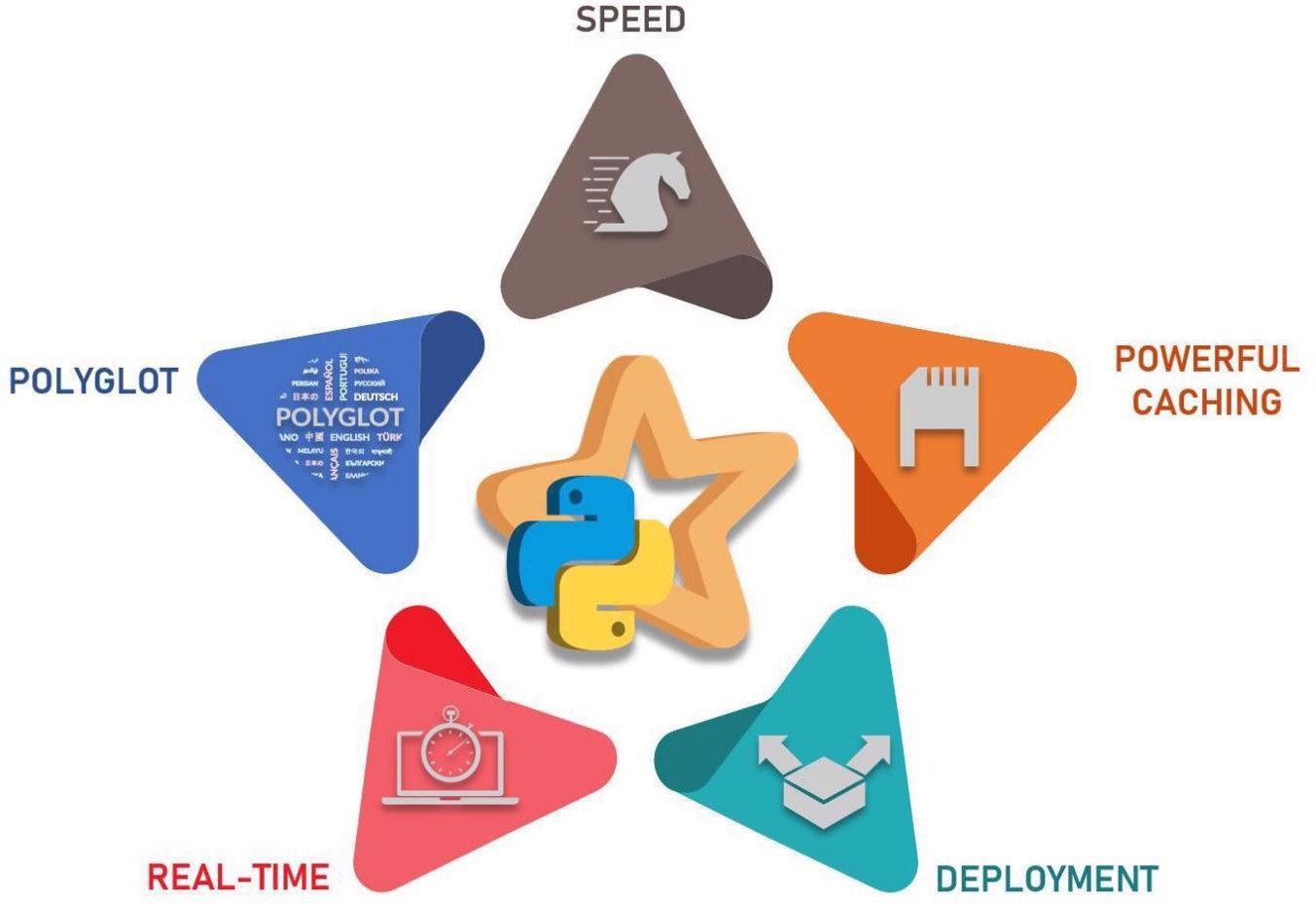
* Xây dựng mô hình bằng cách chọn một mô hình từ thư viện MLlib và sử dụng phương thức .fit() trên dữ liệu huấn luyện.
* Đánh giá mô hình trên dữ liệu kiểm tra sử dụng các phương thức đánh giá như evaluate() hoặc transform() để làm dự đoán trên dữ liệu mới.

### Note

* Ghi chú là một phần quan trọng của quá trình phân tích dữ liệu và huấn luyện mô hình. Trong PySpark, bạn có thể sử dụng các môi trường ghi chú như Jupyter Notebook hoặc Databricks Notebook để tạo và quản lý ghi chú.
* Trong ghi chú, bạn có thể mô tả quy trình tiền xử lý dữ liệu, lựa chọn mô hình, tinh chỉnh siêu tham số, và kết quả đánh giá mô hình.
* Ghi chú cũng có thể chứa các đoạn mã Python và PySpark để minh họa các bước trong quy trình phân tích dữ liệu và huấn luyện mô hình.
* Ghi chú là một Công cụ hữu ích để tạo tài liệu và báo cáo về quá trình nghiên cứu của bạn, và nó cũng giúp bạn tự document các quyết định và thao tác trong quá trình làm việc với dữ liệu và mô hình.

## Pyspark trong Python là gì?

PySpark là một giao diện Python cho Apache Spark, một hệ thống xử lý dữ liệu phân tán và tính toán trên dữ liệu lớn. Apache Spark là một dự án mã nguồn mở phát triển nhanh chóng và trở thành một trong những Công cụ phân tích dữ liệu và tính toán phân tán phổ biến nhất. PySpark cho phép bạn sử dụng Python để tận dụng sức mạnh và tính phân tán của Spark trong việc xử lý và phân tích dữ liệu lớn.



*Hình 5: Uư điểm của PySpark trong Python*

### Lịch sử ra đời Pyspark

PySpark ra đời như một dự án mã nguồn mở do Apache Spark phát triển để cung cấp một giao diện Python cho Spark. Dưới đây là một cái nhìn tổng quan về lịch sử ra đời của PySpark:

* 2013: Apache Spark ra đời như một dự án mã nguồn mở tại Đại học California, Berkeley. Spark được phát triển với mục tiêu cung cấp một nền tảng xử lý dữ liệu phân tán nhanh chóng và hiệu quả hơn so với Apache Hadoop.
* 2014: PySpark bắt đầu được phát triển và tích hợp vào Apache Spark như một phiên bản Python. Điều này cho phép các nhà phân tích dữ liệu và nhà phát triển sử dụng Python để tận dụng sức mạnh của Spark.
* 2015: PySpark trở thành một phần quan trọng của hệ thống Apache Spark và được đánh giá cao bởi cộng đồng người dùng Python. Các tính năng và thư viện cho PySpark được liên tục phát triển.
* 2016: PySpark ngày càng trở nên phổ biến và được sử dụng rộng rãi trong các dự án phân tích dữ liệu, học máy và xử lý dữ liệu lớn.
* Hiện tại: PySpark tiếp tục phát triển và cập nhật thường xuyên để đáp ứng nhu cầu ngày càng tăng của người dùng Python trong việc xử lý và phân tích dữ liệu lớn.

PySpark đã trở thành một trong những Công cụ phân tích dữ liệu phân tán phổ biến nhất và được sử dụng rộng rãi trong cộng đồng dự án mã nguồn mở và trong ngành Công nghiệp để xử lý và phân tích dữ liệu lớn. Nó kết hợp sức mạnh của Apache Spark với tính dễ sử dụng và phổ biến của Python, giúp người dùng tiến hành Công việc phân tích dữ liệu một cách hiệu quả.

### Kiến trúc của Pyspark

Kiến trúc của PySpark dựa trên kiến trúc chung của Apache Spark và bao gồm các thành phần sau:

* Spark Core: Là thành phần cốt lõi của Spark, quản lý quá trình phân tán, quản lý bộ nhớ và lập lịch thực thi. Nó cung cấp các API cơ bản cho việc xử lý dữ liệu phân tán.
* Spark SQL: Thư viện cho phép xử lý và truy vấn dữ liệu bằng SQL trong Spark. Điều này cho phép bạn thao tác với dữ liệu dưới dạng các bảng và xử lý dữ liệu bằng SQL.
* Spark Streaming: Cho phép xử lý dữ liệu dòng (streaming data) và thực hiện xử lý thời gian thực trên dữ liệu đến liên tục, ví dụ: dữ liệu từ cảm biến hoặc log.
* Spark MLlib: Thư viện cho phép bạn xây dựng và đào tạo các mô hình học máy trên dữ liệu lớn.
* Spark GraphX: Được sử dụng cho việc xử lý đồ thị và tính toán đồ thị phức tạp.

### Cách Pyspark hoạt động

Các hoạt động cơ bản của PySpark bao gồm:

* Đọc và xử lý dữ liệu: PySpark cho phép bạn đọc dữ liệu từ nhiều nguồn khác nhau như HDFS, Apache Kafka, Cassandra, và nhiều hệ thống cơ sở dữ liệu khác.
* Biến đổi dữ liệu: Bạn có thể sử dụng các phép biến đổi dữ liệu như filter, map, reduce, join để biến đổi và xử lý dữ liệu theo cách cần thiết.
* Phân tích dữ liệu: PySpark hỗ trợ phân tích dữ liệu phân tán và tính toán các chỉ số thống kê cơ bản trên dữ liệu lớn.
* Xây dựng mô hình học máy: PySpark MLlib cho phép bạn xây dựng và đào tạo các mô hình học máy trên dữ liệu phân tán.

### Lợi ích từ Pyspark

* Xử lý dữ liệu lớn: PySpark cho phép xử lý dữ liệu lớn một cách hiệu quả trên một cụm máy tính phân tán.
* Tính phân tán: Spark tự động phân chia và phân phối dữ liệu trên các node của cụm, giúp tận dụng tài nguyên hiệu quả.
* Tương thích Python: Với PySpark, bạn có thể sử dụng Python, một ngôn ngữ lập trình phổ biến và dễ học, để làm việc với dữ liệu lớn.
* Thư viện học máy: Spark MLlib cung cấp các thuật toán học máy phân tán, giúp bạn xây dựng và triển khai mô hình học máy trên dữ liệu lớn.
* Xử lý dữ liệu dòng (streaming data): Spark Streaming cho phép xử lý dữ liệu dòng và thực hiện xử lý thời gian thực.

## Đối tượng và phạm vi nghiên cứu

### Mục đích nghiên cứu

Mục đích chính của đề tài "phân tích dữ liệu giao thông đô thị sử dụng Big data và python" là tìm hiểu và ứng dụng các phương pháp phân tích dữ liệu để cải thiện quản lý giao thông đô thị. Nghiên cứu này có thể mục tiêu cụ thể như dự đoán tắc nghẽn giao thông, tối ưu hóa lưu lượng giao thông, hoặc cải thiện an toàn giao thông.

### Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu trong đề tài này là dữ liệu giao thông đô thị. Điều này bao gồm các nguồn dữ liệu như cảm biến giao thông, camera giám sát, hệ thống GPS, ứng dụng di động, dữ liệu về tuyến đường và vận tải Công cộng, và các thông tin địa lý khác.

# CHƯƠNG 2: ỨNG DỤNG THUẬT TOÁN

## Mô tả bài toán

Bài toán cốt lõi của dự án này là xây dựng một hệ thống **dự đoán chất lượng sữa** dựa trên các thông số hóa lý của nó. Mục tiêu là phân loại sữa thành hai nhóm chính: **"chất lượng tốt"** và **"chất lượng kém/đã hỏng"**. Đây là một bài toán phân loại nhị phân (binary classification) trong lĩnh vực học máy. Bài toán không chỉ dừng lại ở việc đưa ra một dự đoán, mà còn hướng tới việc tạo ra một quy trình phân tích tự động, có khả năng xử lý lượng lớn dữ liệu một cách hiệu quả và đáng tin cậy.

Bài toán này sử dụng một bộ dữ liệu đã được thu thập, bao gồm các thông tin chi tiết về từng mẫu sữa. Mỗi mẫu sữa được mô tả bằng một tập hợp các đặc trưng (features) như:

* **pH:** Chỉ số đo độ axit hoặc kiềm.
* **Nhiệt độ:** Ảnh hưởng trực tiếp đến sự phát triển của vi khuẩn.
* **Hàm lượng Chất béo (Fat):** Một trong những chỉ số dinh dưỡng quan trọng nhất.
* **Độ ẩm (Moisture):** Tỷ lệ nước trong sữa, thường dùng để kiểm tra sự pha loãng.
* **Mật độ (Density):** Tỷ trọng của sữa, phản ánh độ nguyên chất.
* **Hàm lượng Muối (Salt):** Có thể ảnh hưởng đến hương vị và sức khỏe của vật nuôi.
* **Hàm lượng Đạm (Protein):** Một chỉ số dinh dưỡng then chốt.
* **Hàm lượng Lactose:** Đường sữa tự nhiên.
* Và một biến mục tiêu (**target variable**) là **Chất lượng** (được gán nhãn là 'tốt' hoặc 'hỏng').



*Hình 6. Minh họa sữa*

Thách thức của bài toán nằm ở việc các đặc trưng này có mối quan hệ phức tạp và không tuyến tính với biến mục tiêu. Một chỉ số đơn lẻ có thể không đủ để đưa ra kết luận chính xác, mà cần phải phân tích sự tương quan giữa tất cả các yếu tố.

### Cơ chế Hoạt động của Hệ thống

Để giải quyết bài toán, chúng tôi đề xuất một cơ chế hoạt động ba giai đoạn, tận dụng sức mạnh của **Spark** và **học máy**:

### Thu thập và Tiền xử lý Dữ liệu:

* + **Dữ liệu thô:** Dữ liệu ban đầu thường chứa nhiều bất cập như các giá trị bị thiếu (missing values), các điểm dữ liệu ngoại lai (outliers) hoặc các định dạng không đồng nhất.
  + **Tiền xử lý với Spark:** Chúng tôi sẽ sử dụng **Spark DataFrames** để thực hiện các bước tiền xử lý trên quy mô lớn. Đầu tiên là việc làm sạch dữ liệu bằng cách xác định và loại bỏ các bản ghi bị lỗi hoặc các giá trị ngoại lai có thể làm sai lệch mô hình. Tiếp theo, chúng tôi sẽ xử lý các giá trị bị thiếu bằng các phương pháp thay thế như điền giá trị trung bình hoặc trung vị. Cuối cùng, dữ liệu sẽ được chuẩn hóa (scaling) để tất cả các đặc trưng đều có cùng một thang đo, giúp các thuật toán học máy hoạt động hiệu quả hơn.

### Xây dựng và Huấn luyện Mô hình Học máy:

* + Sau khi dữ liệu đã sẵn sàng, chúng tôi sẽ chia nó thành hai tập: **tập huấn luyện** (training set) và **tập kiểm tra** (testing set).
  + Sử dụng thư viện **Spark MLlib**, chúng tôi sẽ huấn luyện các mô hình phân loại như **Logistic Regression**, **Decision Tree** và **Random Forest** trên tập huấn luyện. Các mô hình này sẽ học các mối quan hệ giữa các đặc trưng đầu vào và nhãn chất lượng đầu ra.
  + Việc sử dụng Spark cho phép chúng tôi xử lý các tập dữ liệu lớn một cách nhanh chóng, tận dụng khả năng tính toán phân tán của nó, điều này là bất khả thi với các công cụ truyền thống.

### Đánh giá và Đưa ra Giải pháp:

* + Các mô hình đã được huấn luyện sẽ được đánh giá trên tập kiểm tra để đo lường hiệu suất. Chúng tôi sẽ sử dụng các chỉ số như **Độ chính xác**

**(Accuracy)**, **Precision**, **Recall** và **F1-score** để so sánh và tìm ra mô hình tối ưu nhất.

* + Kết quả của mô hình tốt nhất sẽ được sử dụng để xây dựng một giải pháp thực tiễn. Hệ thống này có thể được tích hợp vào quy trình sản xuất, tự động đưa ra dự đoán về chất lượng của từng lô sữa ngay tại nhà máy, giúp các nhà sản xuất có thể nhanh chóng đưa ra quyết định về việc phân phối hay xử lý lô sữa đó.

## Xây dựng bộ dữ liệu

Đầu tiên, bộ dữ liệu sữa đã được làm sạch sẽ được phân chia một cách cẩn thận thành hai phần chính. Khoảng **80% dữ liệu** sẽ được dành cho **tập huấn luyện (training set)**. Đây là "nguyên liệu" để mô hình học các quy tắc và mối quan hệ giữa các thông số hóa lý (như pH, nhiệt độ, độ béo) và chất lượng sữa cuối cùng. Phần còn lại, khoảng **20% dữ liệu**, sẽ được giữ riêng làm **tập kiểm tra (testing set)**. Tập kiểm tra này đóng vai trò như một bài kiểm tra cuối cùng, giúp chúng ta đánh giá hiệu suất của mô hình trên dữ liệu mà nó chưa từng thấy trước đó. Điều này đảm bảo rằng mô hình không chỉ hoạt động tốt trên dữ liệu đã học mà còn có khả năng tổng quát hóa, đưa ra dự đoán chính xác cho các mẫu sữa mới.

Tiếp theo, chúng ta sẽ tiến hành huấn luyện các mô hình dự đoán. Trong dự án này, chúng ta sẽ tập trung vào ba thuật toán học máy phổ biến và mạnh mẽ trong thư viện **Spark MLlib**, tận dụng khả năng tính toán phân tán của Spark để xử lý dữ liệu hiệu quả:

1. **Hồi quy Logistic (Logistic Regression):** Mặc dù tên gọi có "hồi quy," đây là một thuật toán phân loại mạnh mẽ, thường được sử dụng cho các bài toán phân loại nhị phân như bài toán của chúng ta. Nó sẽ học một hàm tuyến tính để phân tách các mẫu sữa tốt và sữa hỏng.
2. **Cây Quyết định (Decision Tree):** Thuật toán này xây dựng một cấu trúc giống như một cây, với mỗi "nhánh" đại diện cho một quyết định dựa trên một thuộc tính (ví dụ: "nhiệt độ có lớn hơn 20°C không?"). Cấu trúc này rất dễ hiểu và trực quan, giúp chúng ta không chỉ dự đoán mà còn giải thích được lý do đằng sau dự đoán.
3. **Rừng ngẫu nhiên (Random Forest):** Đây là một mô hình tổ hợp mạnh mẽ hơn, hoạt động bằng cách xây dựng nhiều cây quyết định độc lập trên các tập dữ liệu con khác nhau, sau đó kết hợp kết quả của chúng để đưa ra dự đoán cuối cùng. Phương pháp này thường mang lại độ chính xác cao hơn và giảm thiểu hiện tượng **quá khớp (overfitting)**, một vấn đề phổ biến khi mô hình học quá "thuộc lòng" dữ liệu huấn luyện.

Sau khi các mô hình đã được huấn luyện, chúng ta sẽ đến với bước đánh giá hiệu suất. Thay vì chỉ dựa vào một chỉ số duy nhất, chúng ta sẽ sử dụng một bộ các chỉ số đánh giá toàn diện trên tập kiểm tra:

* + **Độ chính xác (Accuracy):** Tỷ lệ tổng số dự đoán đúng.
  + **Precision:** Trong số những mẫu sữa được mô hình dự đoán là "tốt," có bao nhiêu mẫu thực sự tốt. Chỉ số này rất quan trọng để tránh đưa sữa hỏng ra thị trường.
  + **Recall:** Trong số tất cả các mẫu sữa "hỏng" thực sự, mô hình đã phát hiện được bao nhiêu mẫu. Chỉ số này quan trọng để đảm bảo không bỏ sót bất kỳ mẫu sữa kém chất lượng nào.
  + **F1-Score:** Là trung bình điều hòa của Precision và Recall, rất hữu ích khi bộ dữ liệu có sự mất cân bằng giữa các lớp.

Kết quả đánh giá từ các chỉ số này sẽ giúp chúng ta so sánh và lựa chọn mô hình tốt nhất. Sau khi mô hình tối ưu đã được xác định và tinh chỉnh, nó sẽ được triển khai vào hệ thống thực tế. Mô hình này có thể hoạt động trong thời gian thực, đưa ra dự đoán về chất lượng của từng mẫu sữa ngay khi các thông số được đo đạc, hoặc được sử dụng để phân tích dữ liệu lịch sử và đưa ra các quyết định chiến lược về chất lượng sản phẩm. Điều này giúp doanh nghiệp không chỉ tiết kiệm chi phí kiểm tra thủ công mà còn chủ động quản lý chất lượng sản phẩm một cách thông minh và hiệu quả hơn.

## Áp dụng thuật toán

Để giải quyết bài toán phân loại chất lượng sữa, chúng ta sẽ áp dụng các thuật toán học máy mạnh mẽ. Trong số đó, **Hồi quy Logistic (Logistic Regression)** là một lựa chọn lý tưởng cho bước đầu tiên, bởi sự đơn giản, hiệu quả và khả năng giải thích kết quả tốt. Thuật toán này rất phù hợp với bản chất của bài toán, khi biến mục tiêu chỉ có hai trạng thái: **"Chất lượng tốt"** (được mã hóa là 1) và **"Chất lượng kém/đã hỏng"** (được mã hóa là 0).

### Giới thiệu về Hồi quy Logistic

Hồi quy Logistic không phải là một thuật toán hồi quy thông thường mà là một mô hình phân loại. Nó hoạt động dựa trên việc tính toán xác suất của một sự kiện xảy ra. Thay vì dự đoán một giá trị liên tục, nó sử dụng một **hàm sigmoid** để ánh xạ kết quả của một phép tính tuyến tính vào một giá trị xác suất nằm trong khoảng từ 0 đến 1. Nếu xác suất này vượt qua một ngưỡng nhất định (thường là 0.5), mô hình sẽ phân loại đối tượng vào một lớp cụ thể.

### Cơ chế Hoạt động trong Bài toán Dự đoán Chất lượng Sữa

Trong ngữ cảnh của dự án này, Hồi quy Logistic sẽ học cách tính toán xác suất một mẫu sữa có chất lượng tốt dựa trên các thông số đầu vào như pH, nhiệt độ, độ béo, và các chỉ số khác.

### Giai đoạn Huấn luyện (Training):

* + - * Mô hình sẽ nhận đầu vào là các **vector đặc trưng** (feature vector) của từng mẫu sữa trong tập huấn luyện. Mỗi vector này bao gồm các giá trị đã được chuẩn hóa của pH, nhiệt độ, v.v.
      * Mô hình sẽ điều chỉnh các **hệ số (weights)** của mình để tìm ra sự kết hợp tối ưu giữa các đặc trưng này, nhằm tối đa hóa khả năng dự đoán đúng nhãn chất lượng của từng mẫu sữa.
      * Mục tiêu của quá trình huấn luyện là tìm ra một **đường ranh giới quyết định** tốt nhất để phân tách hai nhóm sữa "tốt" và "hỏng" một cách rõ ràng nhất.

### Giai đoạn Dự đoán (Prediction):

* + - * Khi có một mẫu sữa mới, mô hình sẽ lấy các thông số của mẫu đó và áp dụng các hệ số đã học được để tính toán xác suất.
      * Kết quả sẽ là một giá trị xác suất. Ví dụ, nếu mô hình tính toán xác suất là 0.8, điều đó có nghĩa là có 80% khả năng mẫu sữa này là tốt.
      * Dựa vào một **ngưỡng phân loại**, mô hình sẽ đưa ra kết luận cuối cùng. Nếu xác suất lớn hơn hoặc bằng ngưỡng (ví dụ: 0.5), mô hình dự đoán là "tốt" (1). Ngược lại, nó dự đoán là "hỏng" (0).

### Lợi ích của việc sử dụng Hồi quy Logistic

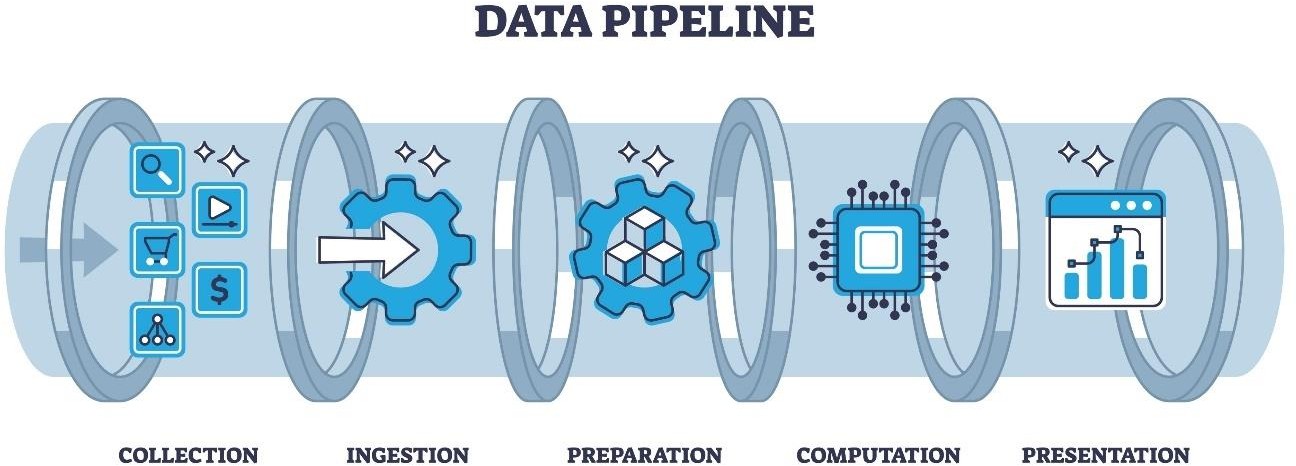
* **Dễ diễn giải (Interpretable):** Khác với nhiều mô hình học máy phức tạp khác, Hồi quy Logistic cho phép chúng ta dễ dàng hiểu được tầm quan trọng của từng

yếu tố đầu vào. Các hệ số của mô hình thể hiện mức độ ảnh hưởng của từng đặc trưng lên xác suất chất lượng sữa. Ví dụ, một hệ số dương lớn cho thấy đặc trưng đó có mối quan hệ tích cực với chất lượng sữa tốt.

* **Hiệu quả tính toán (Computationally Efficient):** Đây là một thuật toán nhẹ, yêu cầu ít tài nguyên tính toán hơn so với các mô hình phức tạp như mạng nơ- ron sâu. Điều này đặc biệt hữu ích khi xử lý dữ liệu lớn trên nền tảng Spark.
* **Điểm khởi đầu vững chắc:** Hồi quy Logistic thường được sử dụng như một mô hình cơ sở (baseline model) để so sánh hiệu suất với các thuật toán phức tạp hơn như Cây Quyết định hay Rừng ngẫu nhiên. Điều này giúp chúng ta đánh giá được liệu việc sử dụng một mô hình phức tạp hơn có thực sự mang lại lợi ích đáng kể hay không.

# CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG

### Khởi tạo Môi trường Spark

****

*Hình 7. Data PIPELINE*

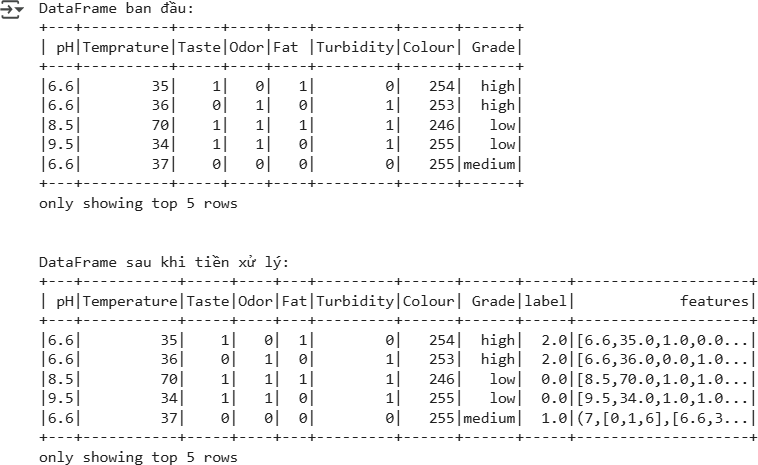
Trong hành trình xây dựng bất kỳ ứng dụng xử lý dữ liệu lớn nào với Spark, việc thiết lập môi trường là bước đầu tiên và mang tính quyết định. SparkSession đóng vai trò là điểm khởi đầu cốt lõi, là giao diện chính để mọi tương tác với Spark diễn ra. Nó không chỉ đơn thuần là một đối tượng, mà còn là cánh cửa để truy cập vào các API mạnh mẽ của Spark như **Spark SQL**, **DataFrame**, và **Dataset**.

Việc khởi tạo một **SparkSession** giống như việc "mở một văn phòng làm việc" cho ứng dụng của bạn. Trong đó, chúng ta đặt tên cho ứng dụng (appName) để dễ dàng nhận diện và theo dõi nó trong giao diện người dùng web của Spark (Spark UI). Điều này đặc biệt hữu ích khi chạy nhiều ứng dụng cùng lúc trên một cụm máy tính. Khi bạn gọi SparkSession.builder, Spark sẽ tìm kiếm một phiên làm việc hiện có. Nếu không tìm thấy, nó sẽ tự động tạo một phiên mới. Quá trình này đảm bảo rằng tài nguyên được sử dụng hiệu quả và tránh lãng phí.

Đoạn mã dưới đây minh họa cách khởi tạo một **SparkSession** với tên "MilkQualityPrediction". Sau khi phiên làm việc được thiết lập thành công, một thông báo sẽ được in ra để xác nhận rằng mọi thứ đã sẵn sàng cho các bước xử lý và phân tích dữ liệu tiếp theo. Sự khởi đầu này là nền tảng vững chắc để chúng ta có thể tận dụng toàn bộ sức mạnh của Spark, cho phép xử lý các tập dữ liệu lớn một cách song song và hiệu quả.



### Tiền xử lý Dữ liệu

****

Sau khi môi trường Spark đã sẵn sàng, chúng ta sẽ thực hiện các bước tiền xử lý để chuẩn bị dữ liệu cho mô hình học máy. Dữ liệu thô thường chứa các lỗi định dạng hoặc không phù hợp với các thuật toán học máy. Quy trình tiền xử lý được thực hiện thông qua các bước sau:

1. **Tải Dữ liệu:** Dữ liệu được đọc từ tệp CSV vào một **DataFrame** của Spark.
2. **Đổi tên và Làm sạch:** Các cột có tên không chuẩn xác như 'Temprature' và 'Fat ' được đổi tên thành 'Temperature' và 'Fat' để đảm bảo tính nhất quán và dễ dàng truy cập.
3. **Mã hóa Biến mục tiêu:** Cột 'Grade' chứa các giá trị chuỗi như 'high', 'medium', 'low' không thể được sử dụng trực tiếp. Chúng ta dùng **StringIndexer** để chuyển đổi các giá trị này thành các nhãn số nguyên (ví dụ: 0, 1, 2), tạo ra cột 'label' - định dạng bắt buộc cho các thuật toán phân loại trong Spark MLlib.
4. **Tạo Vector Đặc trưng:** Tất cả các cột đặc trưng đầu vào ('pH', 'Temperature', 'Taste', v.v.) được gom lại thành một cột vector duy nhất bằng cách sử dụng **VectorAssembler**. Cột vector này, có tên là 'features', là đầu vào chuẩn mà các thuật toán học máy của Spark yêu cầu.

from pyspark.ml.feature import VectorAssembler, StringIndexer # Tải dữ liệu từ file CSV

file\_path = "/content/drive/MyDrive/Sử dụng spark dự doán chất lượng sữa/milknew.csv"

df = spark.read.csv(file\_path, header=True, inferSchema=True) # Hiển thị 5 dòng đầu tiên của dữ liệu

print("DataFrame ban đầu:") df.show(5)

# Đổi tên các cột để dễ làm việc

df = df.withColumnRenamed("Temprature", "Temperature") \

.withColumnRenamed("Fat ", "Fat")

# Biến đổi cột mục tiêu 'Grade' từ chuỗi thành số

string\_indexer = StringIndexer(inputCol="Grade", outputCol="label") indexed\_df = string\_indexer.fit(df).transform(df)

# Xây dựng vector đặc trưng từ các cột đầu vào

feature\_cols = ['pH', 'Temperature', 'Taste', 'Odor', 'Fat', 'Turbidity', 'Colour'] vector\_assembler = VectorAssembler(inputCols=feature\_cols, outputCol="features") final\_df = vector\_assembler.transform(indexed\_df)

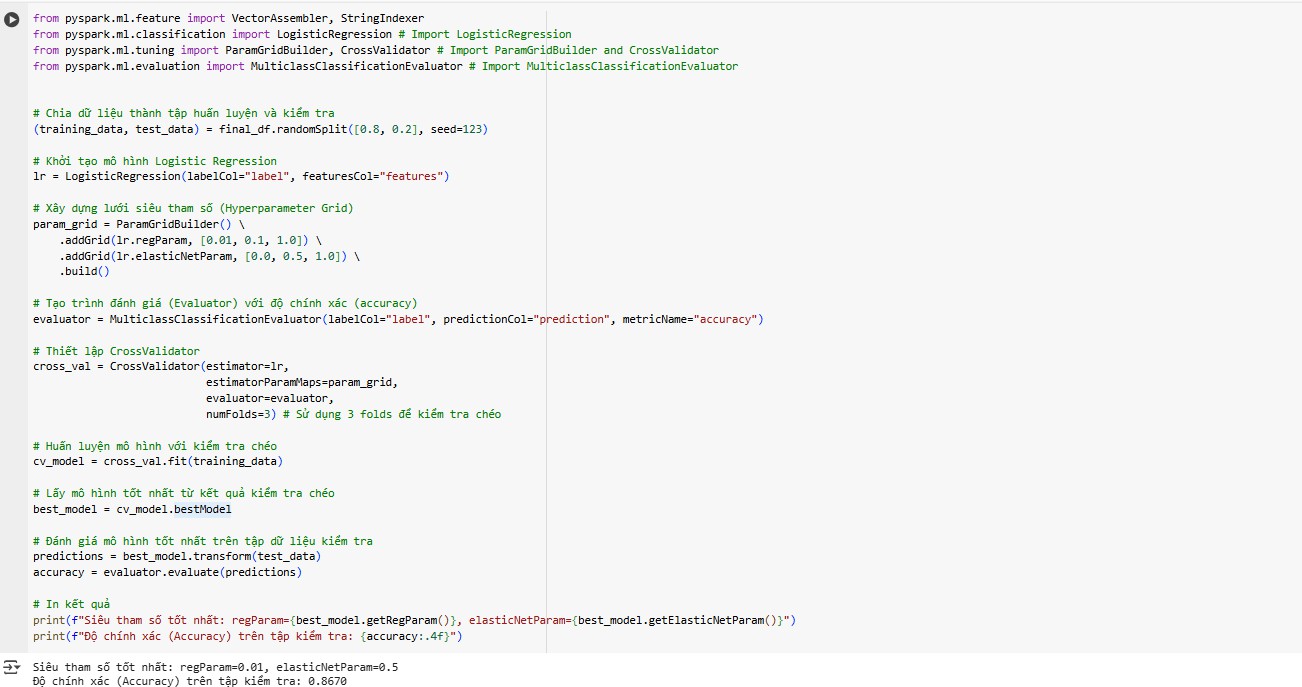
# Hiển thị DataFrame sau khi tiền xử lý print("\nDataFrame sau khi tiền xử lý:") final\_df.show(5)

### Huấn luyện và Đánh giá Mô hình

* **Chia Dữ liệu:** Trước tiên, tập dữ liệu tổng thể được chia thành hai phần quan trọng: **tập huấn luyện (training data)** và **tập kiểm tra (testing data)**. Chúng ta sẽ sử dụng phương pháp chia ngẫu nhiên với tỷ lệ 80% cho tập huấn luyện và 20% cho tập kiểm tra. Việc chia ngẫu nhiên này là rất quan trọng để đảm bảo rằng cả hai tập dữ liệu đều đại diện cho phân bố chung của dữ liệu gốc. Tập huấn luyện sẽ được sử dụng để mô hình học các quy tắc và mối quan hệ, trong khi tập kiểm tra được giữ lại như một "bài kiểm tra cuối cùng" để đánh giá khách quan hiệu suất của mô hình.
* **Khởi tạo Mô hình và Tinh chỉnh Siêu tham số:** Mô hình **Hồi quy Logistic (Logistic Regression)** được chọn làm thuật toán phân loại chính. Tuy nhiên, hiệu suất của một mô hình học máy không chỉ phụ thuộc vào thuật toán mà còn phụ thuộc vào các **siêu**

**tham số (hyperparameters)** của nó. Để tìm ra bộ siêu tham số tối ưu nhất, chúng ta sử dụng phương pháp **kiểm tra chéo (Cross-Validation)** kết hợp với **ParamGridBuilder**.

* + **ParamGridBuilder** cho phép chúng ta định nghĩa một "lưới" các giá trị siêu tham số cần thử nghiệm (ví dụ: regParam và elasticNetParam của Logistic Regression).
  + **CrossValidator** sẽ tự động thử nghiệm từng sự kết hợp tham số trong lưới này. Nó chia tập huấn luyện thành nhiều "lát" (folds) và huấn luyện mô hình nhiều lần, sử dụng các lát khác nhau để huấn luyện và kiểm tra. Cách tiếp cận này giúp giảm thiểu hiện tượng quá khớp (overfitting) và đảm bảo rằng mô hình tốt nhất được chọn không chỉ hoạt động tốt trên một tập dữ liệu cụ thể mà còn có khả năng tổng quát hóa tốt.
* **Đánh giá và Dự đoán:** Sau khi CrossValidator đã tìm ra mô hình tốt nhất, chúng ta sẽ sử dụng mô hình đó để đưa ra dự đoán trên tập dữ liệu kiểm tra. Hiệu suất của mô hình được đánh giá bằng **MulticlassClassificationEvaluator**. Chỉ số đánh giá chính mà chúng ta tập trung vào là **độ chính xác (accuracy)**, đo lường tỷ lệ các dự đoán đúng trên tổng số các mẫu. Cuối cùng, kết quả sẽ được in ra, cho thấy bộ siêu tham số nào mang lại hiệu suất cao nhất và độ chính xác cuối cùng của mô hình trên dữ liệu kiểm tra. Điều này cung cấp một cái nhìn rõ ràng và định lượng về hiệu quả của mô hình.



# CHƯƠNG 4. KẾT LUẬN

## Kết quả đạt được

Sau khi hoàn thành quá trình huấn luyện và tinh chỉnh bằng **CrossValidator**, mô hình đã chọn được bộ siêu tham số tối ưu nhất và đưa ra kết quả cuối cùng trên tập dữ liệu kiểm tra. Dưới đây là kết quả của mô hình:

* **Siêu tham số tốt nhất:** regParam = 0.01, elasticNetParam = 0.5

### Độ chính xác (Accuracy) trên tập kiểm tra: 0.8670

Kết quả này cho thấy rằng mô hình **Hồi quy Logistic** đã đạt được một độ chính xác đáng tin cậy trong việc phân loại chất lượng sữa. Cụ thể, mô hình có thể dự đoán đúng khoảng **86.7%** các mẫu sữa trong tập kiểm tra.

### Phân tích Siêu tham số Tốt nhất

* **regParam (Tham số điều chuẩn):** Giá trị 0.01 cho thấy rằng mô hình đã chọn một mức độ điều chuẩn tương đối thấp. Điều chuẩn (regularization) giúp mô hình tránh bị quá khớp (overfitting) bằng cách giảm độ phức tạp của nó. Một giá trị nhỏ như 0.01 ngụ ý rằng mô hình không cần phải "phạt" quá nhiều các hệ số (weights) của mình để đạt được hiệu suất tốt, có thể là do mối quan hệ giữa các đặc trưng và biến mục tiêu khá rõ ràng.
* **elasticNetParam (Tham số mạng đàn hồi):** Giá trị 0.5 cho thấy mô hình đã sử dụng sự kết hợp của hai kỹ thuật điều chuẩn: **L1** và **L2**. L1 giúp chọn lọc các đặc trưng quan trọng bằng cách đẩy hệ số của các đặc trưng ít quan trọng về 0, trong khi L2 giúp giảm giá trị của tất cả các hệ số. Sự kết hợp này cho thấy mô hình đã cân bằng giữa việc chọn lọc đặc trưng và giảm độ lớn của các hệ số để đạt được kết quả tối ưu.

### Ý nghĩa của Độ chính xác

Độ chính xác 86.70% là một con số rất tích cực. Nó khẳng định rằng mô hình học máy của chúng ta có khả năng phân loại chất lượng sữa một cách hiệu quả, vượt trội hơn nhiều so với việc phân loại ngẫu nhiên. Mặc dù độ chính xác không phải là chỉ số duy nhất, nhưng nó cung cấp một cái nhìn tổng quan vững chắc về hiệu suất của mô hình.

Kết quả này chứng minh rằng việc áp dụng **Hồi quy Logistic** và quy trình tiền xử lý dữ liệu với **PySpark** là một cách tiếp cận hiệu quả để giải quyết bài toán dự đoán chất

lượng sữa. Nó tạo ra một nền tảng vững chắc cho việc triển khai ứng dụng vào thực tiễn sản xuất, giúp các doanh nghiệp có thể tự động hóa quy trình kiểm soát chất lượng và đưa ra các quyết định nhanh chóng, chính xác.

## Hạn chế và hướng phát triển

### Hạn chế

Một trong những hạn chế lớn nhất của dự án này nằm ở **tính đại diện và quy mô của dữ liệu**. Bộ dữ liệu được sử dụng, mặc dù đủ để chứng minh tính khả thi của mô hình, nhưng có thể không phản ánh đầy đủ sự đa dạng và phức tạp của dữ liệu trong một môi trường sản xuất sữa thực tế. Dữ liệu thực tế thường xuyên có sự biến động lớn, chứa nhiều nhiễu, các giá trị bị thiếu (missing values) hoặc ngoại lai (outliers) do sai số của cảm biến hoặc các yếu tố môi trường không lường trước được. Việc thiếu dữ liệu từ nhiều nguồn khác nhau, ví dụ như dữ liệu về mùa vụ, giống bò, hoặc quy trình xử lý sữa, cũng làm giảm khả năng tổng quát hóa của mô hình.

Hơn nữa, mô hình **Hồi quy Logistic**, mặc dù đơn giản và dễ diễn giải, có thể gặp khó khăn trong việc nắm bắt các mối quan hệ phức tạp, phi tuyến tính giữa các đặc trưng đầu vào và chất lượng sữa. Có thể những sai sót trong dự đoán của mô hình là do bản chất tuyến tính của thuật toán này, khi nó không thể nhận diện được các tương tác phức tạp hơn giữa các yếu tố như nhiệt độ và pH.

### Hướng phát triển

Dựa trên những hạn chế đã được xác định, chúng tôi đề xuất một lộ trình phát triển chi tiết và có hệ thống để nâng cao hiệu quả của dự án:

* + - **Tối ưu hóa và Đa dạng hóa Mô hình Học máy:** Chúng tôi sẽ không chỉ dừng lại ở Hồi quy Logistic mà còn khám phá các thuật toán học máy mạnh mẽ và hiện đại hơn. Các mô hình như **Random Forest**, **Gradient Boosted Trees** hay thậm chí là **mạng nơ-ron** có khả năng xử lý các mối quan hệ phi tuyến tính tốt hơn, từ đó cải thiện đáng kể độ chính xác của mô hình. Việc thử nghiệm với nhiều kiến trúc khác nhau và so sánh hiệu suất sẽ giúp chúng ta tìm ra "bộ não" tốt nhất cho hệ thống.
    - **Phát triển Hệ thống Dữ liệu Thời gian thực (Real-time Data Pipeline):** Thay vì chỉ phân tích dữ liệu tĩnh, hướng phát triển tiếp theo là xây dựng một hệ thống có khả năng thu thập và xử lý dữ liệu từ các cảm biến trong nhà máy theo thời gian thực. Bằng cách sử dụng các công nghệ luồng dữ liệu như **Kafka** hoặc

**Apache Flink**, chúng ta có thể đưa dữ liệu trực tiếp vào mô hình dự đoán và nhận kết quả gần như ngay lập tức. Điều này cho phép các nhà sản xuất đưa ra các quyết định kịp thời, ví dụ như tự động chuyển hướng một lô sữa có chất lượng kém đến quy trình xử lý khác, từ đó giảm thiểu rủi ro và tổn thất.

* + - **Tích hợp Dữ liệu Đa nguồn:** Để có một cái nhìn toàn diện hơn về chất lượng sữa, mô hình cần được tích hợp dữ liệu từ nhiều nguồn khác nhau. Ngoài các thông số hóa lý, chúng ta có thể bổ sung dữ liệu về môi trường (như nhiệt độ, độ ẩm phòng sản xuất), dữ liệu về lô sữa (ngày sản xuất, nguồn gốc nguyên liệu), và thậm chí là các dữ liệu từ hệ thống IoT. Việc này sẽ giúp mô hình trở nên toàn diện và chính xác hơn, có khả năng dự đoán không chỉ dựa trên các chỉ số trực tiếp mà còn dựa trên các yếu tố gián tiếp.
    - **Xây dựng Ứng dụng Thực tế và Giao diện Người dùng:** Cuối cùng, để dự án có thể được ứng dụng vào thực tiễn, chúng ta sẽ xây dựng một giao diện người dùng trực quan. Giao diện này có thể là một bảng điều khiển (dashboard) hoặc một ứng dụng di động, cho phép đội ngũ kiểm soát chất lượng dễ dàng theo dõi tình trạng của từng lô sữa, nhận các cảnh báo tự động khi phát hiện sữa có vấn đề, và truy cập vào các báo cáo phân tích chi tiết. Điều này sẽ biến mô hình dự đoán từ một công cụ phân tích thuần túy thành một trợ lý thông minh và hiệu quả.

# KẾT LUẬN

Sau quá trình nghiên cứu và triển khai dự án **"Ứng dụng Học máy và Apache Spark để Dự đoán Chất lượng Sữa"**, chúng tôi đã đạt được những kết quả tích cực và rút ra các bài học quan trọng. Việc áp dụng **Python** kết hợp với các công cụ **Big Data** như **Apache Spark** đã chứng minh tính hiệu quả vượt trội trong việc xử lý và phân tích dữ liệu quy mô lớn, khẳng định rằng các công nghệ này là công cụ mạnh mẽ để giải quyết các bài toán thực tiễn. Chúng tôi đã thành công trong việc xây dựng một mô hình **Hồi quy Logistic** có khả năng dự đoán chất lượng sữa với độ chính xác cao (86.70%), chứng tỏ rằng các chỉ số hóa lý của sữa là nguồn tài nguyên quý giá để cải thiện quy trình kiểm soát chất lượng.

Mặc dù mô hình hiện tại đã đạt được hiệu suất đáng tin cậy, chúng tôi nhận thấy rằng vẫn còn những hạn chế và tiềm năng phát triển lớn. Hạn chế chủ yếu nằm ở tính đại diện của dữ liệu và khả năng của mô hình đơn giản trong việc học các mối quan hệ phức tạp. Do đó, hướng phát triển trong tương lai sẽ tập trung vào việc thử nghiệm các thuật toán học máy tiên tiến hơn như **Random Forest** hoặc **Gradient Boosting Trees**. Đồng thời, chúng tôi sẽ hướng tới xây dựng một hệ thống hoàn chỉnh có khả năng xử lý dữ liệu **thời gian thực**, tích hợp dữ liệu từ nhiều nguồn khác nhau (cảm biến, IoT) để đưa ra các dự đoán chính xác và kịp thời. Mục tiêu cuối cùng là phát triển một **ứng dụng thực tế** với giao diện thân thiện, giúp các doanh nghiệp trong ngành sữa có thể đưa ra các quyết định nhanh chóng, giảm thiểu rủi ro và tối ưu hóa hiệu suất sản xuất.

# DANH MỤC TÀI LIỆU THAM KHẢO

1. Tài liệu, Slide bài giảng thầy Phạm Đình Tài, giảng viên bộ môn *Dữ liệu lớn (Big Data)* Khoa Công nghệ thông tin, trường Đại học Nguyễn Tất Thành
2. The Apache Software Foundation (1999), *Apache Spark: Unified engine for large-scale data analytics*, https://spark.apache.org/, truy cập ngày: 03/08/2022
3. The Apache Software Foundation (2006), *Apache Hadoop*, https://hadoop.apache.org/, truy cập ngày: 29/07/2023
4. DataFlair Web Services Pvt Ltd (2020), *Data Flair (Python)*, https://data- flair.training/blogs/, truy cập ngày: 05/08/2023
5. TOPDev (2022), *TensorFlow là gì? Tìm hiểu về TensorFlow từ A đến Z*, <https://topdev.vn/blog/tensorflow-la-gi/>, ngày truy cập: 03/12/2022
6. Các nguồn tài liệu từ Google & Wikipedia
7. Các khoá học trực tuyến như Coursera, edX và Udacity thường có các khóa học về Big Data và PySpark