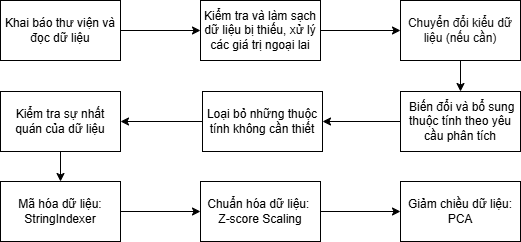
Bước tiền xử lý dữ liệu bằng pypark

1. **Tổng quan các bước tiền xử lý**



Hình…:Sơ đồ các bước tiền xử lý dữ liệu

1. Khai báo thư viện

Khai báo thư viện đồng thời bắt đầu 1 spark session

A screen shot of a computer

Description automatically generated

1. Đọc và xem mô tả, thống kê dữ liệu, kiểm tra xử lý dữ liệu trống (NULL)

Import dữ liệu và đọc dữ liệu *marketing\_campaign.csv* bằng *spark.read.csv*



Xem dữ liệu

A screen shot of a computer

Description automatically generated

Khai báo hàm col, count, isnan, when của thư viện *pyspark.sql.function* để đếm số dữ liệu không Null ở từng cột. Đồng thời xem lại các kiểu dữ liệu của từng cột bằng *printSchema()* đã hợp lý chưa.

A screen shot of a computer

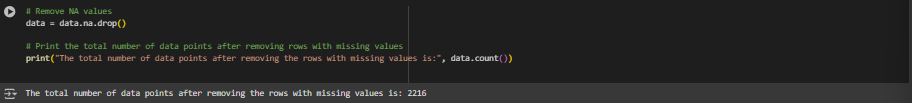
Description automatically generated

A computer screen shot of a person

Description automatically generated



Ta thấy ở cột “Income” bị giá trị null. Có nhiều cách xử lý nhưng ta sẽ bỏ các hàng.



1. Chuyển đổi kiểu dữ liệu, biến đổi thuộc tính thuộc tính

Ta còn phát hiện ở Dt\_Customer là kiểu dữ liệu string. Nhưng đây là ngày khách hàng đăng ký với công ty. Vì thế ta sẽ thay đổi kiểu dữ liệu từ *string* sang *timestamp*

Tạo một cột *CustomerFor* là xem ngày gắn bó của khách hàng với cửa hàng. Vì đây là dữ liệu quá khứ nên ta sẽ sử dụng thời gian gần nhất hiện tại làm mốc để xem. Do kiểu dữ liệu của số ngày đăng ký mua hàng đầu tiên *Dt\_Customer* là kiểu *timestamp* nên ta sẽ dùng hàm *datediff* để đếm số ngày sau đó ép kiểu nó sang Interger.

A screenshot of a computer

Description automatically generated

Để thuận tiện cho việc phân loại. Chúng ta sẽ thay đổi 1 số biến và loại bỏ một số biến không cần thiết.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Thêm cột Is\_Parent với 2 giá trị 0 ,1 (Với 0 là không phải là bố hoặc mẹ và ngược lại) ta có thể đây là 1 biến phân loại.

Đổi tên 2 biến *Education\_index*  và *Living\_With\_index* để ngắn gọn hơn.



1. Phân tích một số thuộc tính và kiểm tra sự nhát quán của dữ liệu.

Tiếp tục sử scatter plot đế xem sự phân bố sau khi đã dọn sạch dữ liệu

A computer screen shot of a black screen

Description automatically generated

Kết quả:

A screenshot of a graph

Description automatically generated

Ta có thể nhận thế một số cột quan trọng có oulier là *Age* và *Icome*. Vì Ta tính độ tuổi từ năm hiện tại là 2024 nên sẽ xuất hiện những outlier có độ tuổi quá 100 tuổi.

Dựa theo sự phân bố. Ta áp dụng điều kiện là “Age <90” và “Income<600000” để loại bỏ outlier

A black screen with white text

Description automatically generated

Sau khi đã dọn dữ liệu. Ta thực hiện các bước chuẩn bị trước khi thực hiện K-means model.

1. Chuyển đổi kiểu dữ liệu string sang interger

Những biến phân loại có kiểu dữ liệu string và giá trị dữ liệu mà tên hay cấp bậc ta sẽ chuyển chúng thành số tất cả.

A screenshot of a computer

Description automatically generated

Kết quả

A screenshot of a computer

Description automatically generated

1. Chuẩn hóa dữ liệu

Loại bỏ một cột là những giá trị khuyến mãi. Vì những thuộc tính không liên quan đến những đặc tính cá nhân của khách hàng vì thế chúng ta loại bỏ đi để chuẩn hóa dữ liệu.

A screenshot of a computer

Description automatically generated

Sau khi đã xong ta tải dữ liệu này về để khi thực hiện model k-means dùng bộ dữ liệu này để có đưa ra được đánh giá.



Tiếp theo ta chuẩn hóa tất cả các dữ liệu dựa trên công thức của thư viện skicit-learn



Tạo một dictionary để lưu giá trị trung bình và độ lệch chuẩn của từng cột

A screen shot of a computer

Description automatically generated

Sau đó chạy thử công thức

A black and grey background with colorful lines

Description automatically generated with medium confidence

Sau đó dùng hàm *for* để chuẩn hóa cả bộ dữ liệu

A screen shot of a computer

Description automatically generated

1. Xuất dữ liệu sau khi tiền xử lý dữ liệu

