ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

Môn học: Công nghệ dữ liệu lớn

Phân tích giao dịch thẻ tín dụng giả mạo

Giảng viên hướng dẫn: Ts. Đỗ Trọng Hợp

Sinh viên thực hiện:		
STT	**Họ tên**	**MSSV**
1	Nguyễn Trung Hiếu	22550004
2	Ngô Trần Tuấn Phong	22550016
3	Đào Nhâm Phúc	22550017
4	Phạm Hoàng Sang	22550019

Thành phố Hồ Chí Minh – 05/2024

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập - Tự Do - Hạnh Phúc

TP. HCM, ngày 01 tháng 05 năm 2024

NHẬN XÉT ĐỒ ÁN MÔN HỌC

Tên đồ án:

Phân tích giao dịch thẻ tín dụng giả mạo

Nhóm SV thực hiện:

Họ và tên: Nguyễn Trung Hiếu

MSSV: 22550004

Họ và tên: Ngô Trần Tuấn Phong

MSSV: 22550016

Họ và tên: Đào Nhâm Phúc

MSSV: 22550017

Họ và tên: Phạm Hoàng Sang

MSSV: 22550019

Giảng viên phụ trách:

Ts. Đỗ Trọng Hợp

Đánh giá Đồ án:

1. Về cuốn báo cáo:

Số trang : Số chương :
Số bảng số liệu : Số hình vẽ :
Số tài liệu tham khảo : Sản phẩm :

Một số nhận xét về hình thức cuốn báo cáo:

---

2. Về nội dung nghiên cứu:

---

3. Về thái độ làm việc của sinh viên:

Đánh giá chung:

Điểm sinh viên:

Nguyễn Trung Hiếu:……../10

Ngô Trần Tuấn Phong:……../10

Đào Nhâm Phúc:……../10

Phạm Hoàng Sang:……../10

Người nhận xét
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Trải qua thời gian học tập và rèn luyện cùng Thầy và các bạn, bản thân chúng em đã tiếp thu được rất nhiều những kiến thức.

Chúng em xin chân thành cảm ơn Thầy Đỗ Trọng Hợp đã tận tình giảng dạy, hướng dẫn và giúp đỡ chúng em trong suốt quá trình học tập.

Cuối cùng, chúng em chúc Thầy công tác tốt và luôn dồi dào sức khỏe để có thể cống hiến hết mình cho những thế hệ sinh viên tiếp theo.

Chúng em xin chân thành cảm ơn!

Sinh viên thực hiện

Nguyễn Trung Hiếu

Ngô Trần Tuấn Phong

Đào Nhâm Phúc

Phạm Hoàng Sang

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập - Tự Do - Hạnh Phúc

ĐỀ CƯƠNG CHI TIẾT

TÊN ĐỒ ÁN:

Phân tích giao dịch thẻ tín dụng giả mạo

Cán bộ hướng dẫn:
Đỗ Trọng Hợp

Thời gian thực hiện:

Từ ngày 01-05-2024 đến ngày 10-06-2024

Sinh viên thực hiện:

Nguyễn Trung Hiếu – 22550004

Ngô Trần Tuấn Phong – 22550016

Đào Nhâm Phúc – 22550017

Phạm Hoàng Sang – 22550019

Nội dung đồ án:

- Giới thiệu về gian lận thẻ tín dụng
- Giới thiệu về kiến trúc hệ thống Streaming
- Giới thiệu về công nghệ sử dụng
- Quy trình xử lý
- Phân tích dữ liệu và đặc trưng
- Ứng dụng vào hệ thống
- Kết quả hệ thống
- Hạn chết và hướng phát triển
- Kết luận

Kế hoạch thực hiện:

Tìm hiểu và demo về phân tích giao dịch thẻ tín dụng giả mạo

Xác nhận của Giảng viên	**TP. HCM, ngày 10 tháng 06 năm 2024**
(Ký tên và ghi rõ họ tên)	**Sinh viên** (Ký tên và ghi rõ họ tên)

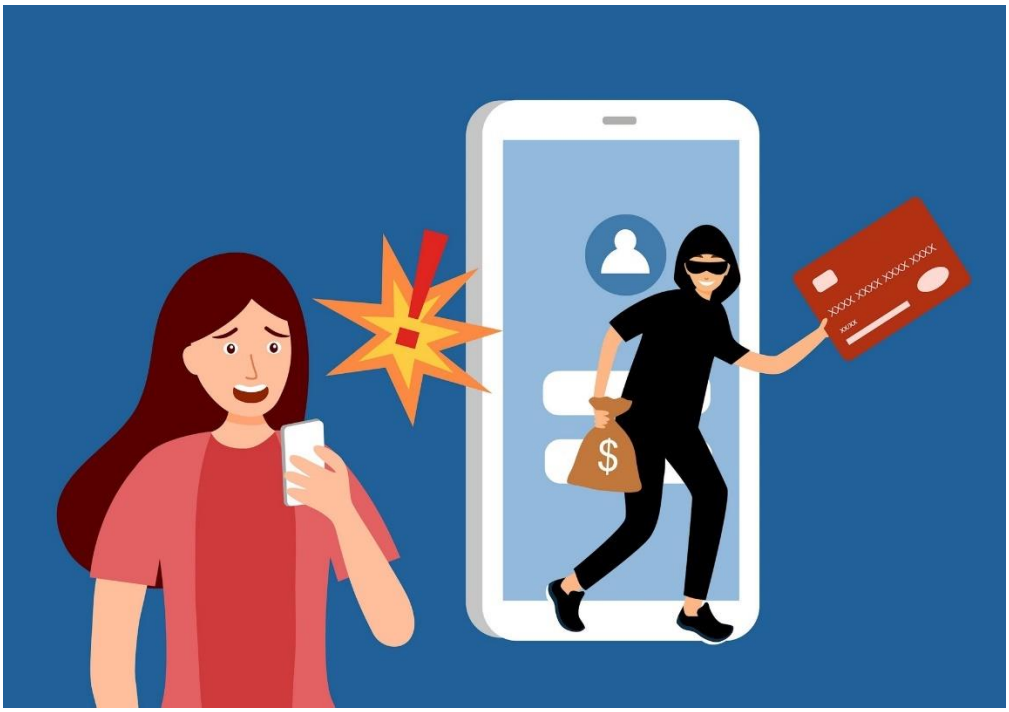
	Nguyễn Trung Hiếu, Ngô Trần Tuấn Phong, Đào Nhâm Phúc, Phạm Hoàng Sang

MỤC LỤC

DANH MỤC HÌNH ẢNH

I. Giới thiệu về gian lận thẻ tín dụng

Gian lận thẻ tín dụng là một vấn đề nghiêm trọng gây thiệt hại lớn cho các tổ chức tài chính và người tiêu dùng. Việc phát hiện sớm các giao dịch gian lận là rất quan trọng để giảm thiểu tổn thất. Báo cáo này trình bày việc phân tích giao dịch thẻ tín dụng giả mạo sử dụng Kafka, S park Streaming và thuật toán Random Forest Classifier trong PySpark. Hệ thống được thiết kế để xử lý một lượng lớn dữ liệu giao dịch thẻ tín dụng theo thời gian thực, phân tích các mẫu g iao dịch và xác định các giao dịch đáng ngờ.



Hình 1: Gian lận thẻ tín dụng

II. Giới thiệu về kiến trúc hệ thống Streaming

Hệ thống phân tích giao dịch thẻ tín dụng giả mạo được thiết kế theo kiến trúc Streaming, cho phép xử lý dữ liệu theo thời gian thực và phản hồi nhanh chóng trước các hoạt động gian lận. Hệ thống bao gồm các thành phần chính sau:

1. Thu thập dữ liệu

1.1 Nguồn dữ liệu

Dữ liệu giao dịch thẻ tín dụng được thu thập từ nhiều nguồn khác nhau, bao gồm:

- Hệ thống thanh toán trực tuyến (ví dụ: website bán hàng, ứng dụng di động)
- Điểm bán hàng (POS) tại các cửa hàng
- Máy ATM
- Hệ thống chuyển khoản ngân hàng

- Các nguồn dữ liệu khác liên quan đến hoạt động giao dịch thẻ tín dụng.

1.2 Phương thức thu thập

- Dữ liệu được thu thập theo thời gian thực, sử dụng các API hoặc các giao thức truyền thông khác.
- Sử dụng các công cụ logging để ghi lại thông tin giao dịch.
- Sử dụng các dịch vụ tích hợp (integration services) để kết nối với các nguồn dữ liệu khác nhau.

2. Lưu trữ và truyền dữ liệu

- **Apache Kafka:** Nền tảng streaming phân tán được sử dụng để lưu trữ và truyền dữ liệu giao dịch thẻ tín dụng theo thời gian thực. Kafka hoạt động như một hệ thống message queue, cho phép các ứng dụng khác nhau gửi và nhận dữ liệu một cách hiệu quả.
    - Kafka đảm bảo dữ liệu được truyền tải một cách tin cậy và nhanh chóng, với kh ả năng mở rộng để đáp ứng nhu cầu xử lý dữ liệu lớn.
    - Kafka được thiết kế để xử lý một lượng lớn dữ liệu với độ trễ thấp, phù hợp với xử lý dữ liệu stream trong thời gian thực.

3. Xử lý dữ liệu

- **Apache Spark Streaming**: Khung xử lý dữ liệu stream hiệu quả cao, được dụng để xử lý dữ liệu từ Kafka. Spark Streaming chia dữ liệu stream thành các micro-batch, sau đó xử lí thành micro-batch bằng cách sử dụng các phép biến đổi Spark
    - Spark Streaming cung cấp khả năng xử lý dữ liệu theo thời gian thực, với tốc độ cao và khả năng mở rộng.
    - Framework này hỗ trợ nhiều phép biến đổi, cho phép thực hiện các tác vụ như lọc, ánh xạ, reduce, join và aggregation.

4. Phân tích và phân loại

- **PySpark:** API Python cho Spark, được sử dụng để xây dựng model Machine Learning và phân tích dữ liệu. PySpark cung cấp một giao diện Pythonic đơn giản và dễ sử dụng, cho phép các nhà phát triển Python dễ dàng làm việc với Spark.
- **Random Forest Classifier:** Thuật toán Machine Learning được sử dụng để phân loại giao dịch giả mạo và hợp lệ. Random Forest Classifier kết hợp nhiều cây quyết định để tạo ra một model mạnh mẽ hơn, có độ chính xác cao và khả năng chống overfitting.

- Model Random Forest được huấn luyện bằng dữ liệu lịch sử, sau đó được sử dụng để phân loại các giao dịch mới theo thời gian thực.

5. **Cảnh báo và phản hồi**
   - **Hệ thống cảnh báo:** Hệ thống cảnh báo được tích hợp để thông báo cho người dùng về các giao dịch được phân loại là giả mạo. Cảnh báo có thể được gửi qua email, tin nhắn SMS hoặc các hệ thống thông báo khác.
   - **Phản hồi:** Hệ thống phản hồi được thiết kế để thực hiện các hành động cần thiết khi phát hiện giao dịch giả mạo. Ví dụ:
     - Từ chối giao dịch
     - Khóa thẻ tín dụng
     - Liên lạc với chủ thẻ để xác minh giao dịch
     - Báo cáo vụ gian lận cho cơ quan chức năng.
6. **Quản lý và giám sát**
   - **Hệ thống quản lý:** Hệ thống quản lý được sử dụng để giám sát hoạt động của hệ thống, theo dõi hiệu suất, quản lý model và cập nhật dữ liệu huấn luyện.
   - **Công cụ giám sát:** Hệ thống sử dụng các công cụ giám sát để theo dõi các chỉ số quan trọng như độ chính xác của model, số lượng giao dịch giả mạo được phát hiện, thời gian xử lý dữ liệu.

III. Giới thiệu về công nghệ sử dụng

1. Apache Kafka

Apache Kafka là một nền tảng stream phân tán, được thiết kế để xử lý một lượng lớn dữ liệu theo thời gian thực. Kafka hoạt động theo mô hình publish-subscribe, cho phép các ứng dụng khác nhau gửi và nhận dữ liệu một cách hiệu quả.

- **Kiến trúc:** Kafka được xây dựng dựa trên kiến trúc phân tán, bao gồm các thành phần chính sau:
  - **Broker:** Mỗi Kafka cluster bao gồm một hoặc nhiều broker, chịu trách nhiệm lưu trữ dữ liệu và xử lý các yêu cầu từ producer và consumer.
  - **Topic:** Dữ liệu trong Kafka được tổ chức thành các topic. Mỗi topic là một stream dữ liệu được chia thành các partition.
  - **Partition:** Mỗi topic được chia thành một hoặc nhiều partition, cho phép phân tán dữ liệu trên nhiều broker và tăng cường khả năng mở rộng.

- **Producer:** Các ứng dụng gửi dữ liệu vào Kafka được gọi là producer. Producer gửi dữ liệu đến các topic cụ thể.
- **Consumer:** Các ứng dụng đọc dữ liệu từ Kafka được gọi là consumer. Comsumer đăng ký vào các topic cụ thể và nhận dữ liệu từ các partition của topic đó.

- **Ưu điểm của Kafka:**
  - **Khả năng mở rộng:** Kafka được thiết kế để xử lý một lượng lớn dữ liệu theo thời gian thực và có thể mở rộng để đáp ứng nhu cầu ngày càng tăng.
  - **Độ tin cậy:** Kafka sử dụng cơ chế replication để đảm bảo dữ liệu được sao lưu và khả năng chịu lỗi cao
  - **Hiệu suất:** Kafka có khả năng xử lý dữ liệu với tốc độ cao, cho phép các ứng dụng phản hồi nhanh chóng với các sự kiện theo thời gian thực.
  - **Độ trễ thấp:** Kafka có khả năng cung cấp dữ liệu với độ trễ thấp, cho phép các ứng dụng phản hồi nhanh chóng với các sự kiện mới nhất.

2. Apache Spark Streaming

Apache Spark Streaming là một framework xử lý dữ liệu stream được xây dựng trên nền tảng Apache Spark. Spark Streaming cho phép xử lý dữ liệu theo thời gian thực từ các nguồn dữ liệu khác nhau, bao gồm Kafka.

➢ **Cách thức hoạt động:** Spark Streaming chia dữ liệu stream thành các micro-batch, sau đó xử lý từng micro-batch bằng cách sử dụng các phép biến đổi Spark.

➢ **DStreams:** Spark Streaming sử dụng khái niệm DStream (Discretized Stream) để biểu diễn dữ liệu stream. DStream là một chuỗi các RDD (Resilient Distributed Dataset), mỗi RDD đại diện cho dữ liệu trong một khoảng thời gian cụ thể.

➢ **Phép biến đổi:** Spark Streaming hỗ trợ nhiều phép biến đổi trên DStream, cho phép thực hiện các tác vụ như lọc, ánh xạ, reduce, join và aggregation.

➢ **Kết quả đầu ra:** Kết quả của xử lý Spark Streaming có thể được ghi vào các hệ thống lưu trữ khác nhau, bao gồm cơ sở dữ liệu, hệ thống file hoặc các nền tảng stream khác.

3. PySpark

Hình 2: PySpark

PySpark là một công cụ mạnh mẽ cho phép bạn tận dụng sức mạnh của Apache Spark trong thế giới Python, mang đến giải pháp hiệu quả cho việc xử lý dữ liệu quy mô lớn. Được phát triển bởi Apache Software Foundation, PySpark kết hợp khả năng xử lý song song và phân tán của Spark với sự dễ dàng sử dụng và linh hoạt của Python, biến nó thành một công cụ lý tưởng cho nhiều nhiệm vụ liên quan đến dữ liệu, bao gồm:

3.1 Xử lý dữ liệu quy mô lớn

PySpark là giải pháp hoàn hảo cho việc xử lý các tập dữ liệu lớn mà máy tính cá nhân không thể xử lý hiệu quả. Sử dụng tính năng phân tán của Spark, PySpark chia tập dữ liệu thành các phần nhỏ, phân bổ chúng cho nhiều node trong cụm, cho phép xử lý đồng thời và tăng tốc độ tính toán đáng kể

3.2 Phân tích dữ liệu

PySpark cung cấp các công cụ hiệu quả để phân tích dữ liệu, bao gồm Spark SQL, một engine truy vấn SQL dựa trên Spark, cho phép bạn truy vấn dữ liệu hiệu quả và nhanh chóng. PySpark cũng hỗ trợ các thao tác xử lý dữ liệu thông qua RDD (Resilient Distributed Dataset) và DataFrame, cho phép bạn thực hiện các phép biến đổi phức tạp trên dữ liệu một cách dễ dàng

3.3 Machine Learning

PySpark cung cấp thư viện MLlib, một bộ công cụ học máy mạnh mẽ tích hợp sẵn trong Spark, cho phép bạn xây dựng và đào tạo các model học máy hiệu quả trên dữ liệu quy mô lớn. PySpark hỗ trợ một loạt các thuật toán học máy, bao gồm hồi quy tuyến tính, phân loại, cụm và khuyến nghị.

3.4 Xử lý dữ liệu Stream

PySpark Streaming là một mô-đun mạnh mẽ cho phép bạn xử lý dữ liệu streaming theo thời gian thực. Bạn có thể sử dụng PySpark Streaming để phân tích dữ liệu từ các nguồn như Kafka, Twitter và các ứng dụng tạo ra dữ liệu liên tục.

3.5 Xử lý dữ liệu đồ thị

PySpark GraphX là một mô-đun cung cấp các công cụ để xử lý dữ liệu đồ thị trên Spark. Bạn có thể sử dụng PySpark GraphX để phân tích các mạng lưới phức tạp, chẳng hạn như mạng xã hội, mạng truyền thông và các mạng sinh học.

Ưu điểm chính của PySpark:

- **Dễ sử dụng:** PySpark cung cấp một giao diện thân thiện với Python, giúp các nhà phát triển Python dễ dàng chuyển sang sử dụng Spark.
- **Hiệu suất cao:** PySpark tận dụng sức mạnh của Spark để xử lý dữ liệu lớn với tốc độ cao và hiệu quả.
- **Linh hoạt:** PySpark hỗ trợ nhiều thư viện Python phổ biến, cho phép tích hợp dễ dàng với các hệ thống và công cụ khác.
- **Hỗ trợ cộng đồng mạnh mẽ:** PySpark có một cộng đồng người dùng lớn và tích cực, cung cấp hỗ trợ và tài liệu phong phú.
- **Khả năng mở rộng:** PySpark có thể mở rộng để xử lý các tập dữ liệu lớn hơn và xử lý nhiều dữ liệu hơn bằng cách thêm nhiều node vào cụm Spark.

Một số ví dụ về việc sử dụng PySpark:

- **Xây dựng một hệ thống phân tích dữ liệu web:** Bạn có thể sử dụng PySpark để thu thập và phân tích dữ liệu từ các trang web, chẳng hạn như nhật ký truy cập và dữ liệu người dùng.
- **Tạo các model dự đoán cho thị trường chứng khoán:** PySpark có thể được sử dụng để huấn luyện các model học máy để dự đoán giá cổ phiếu và các chỉ số tài chính khác.
- **Phát hiện gian lận:** PySpark có thể được sử dụng để phát hiện gian lận thẻ tín dụng, gian lận bảo hiểm và các hình thức gian lận khác.
- **Xây dựng các hệ thống recommend:** PySpark có thể được sử dụng để xây dựng các hệ thống khuyến nghị, chẳng hạn như khuyến nghị phim, sản phẩm và âm nhạc.
- **Phân tích dữ liệu mạng xã hội:** PySpark có thể được sử dụng để phân tích dữ liệu mạng xã hội, chẳng hạn như xu hướng chủ đề, cảm xúc và tương tác người dùng.

Tóm lại, PySpark là một công cụ mạnh mẽ cho phép bạn tận dụng sức mạnh của Apache Spark trong thế giới Python, mang đến giải pháp hiệu quả cho việc xử lý dữ liệu quy mô lớn, phân tích dữ liệu, học máy và xử lý dữ liệu stream.

4. Random Forest Classifier

Random Forest Classifier là một thuật toán Machine Learning thuộc nhóm ensemble learning. Thuật toán này kết hợp nhiều cây quyết định (decision tree) để tạo ra một model mạnh mẽ hơn.

- **Cách thức hoạt động:**
    1. **Bagging:** Thuật toán tạo ra nhiều tập dữ liệu con bằng cách lấy mẫu ngẫu nhiên từ tập dữ liệu ban đầu.
    2. **Tạo cây quyết định:** Một cây quyết định được xây dựng cho mỗi tập dữ liệu con
    3. **Kết hợp:** Các cây quyết định được kết hợp để tạo ra một model duy nhất
- **Ưu điểm:**
    - **Độ chính xác cao:** Random Forest thường có độ chính xác cao hơn so với các thuật toán cây quyết định đơn lẻ.
    - **Khả năng xử lý dữ liệu phức tạp:** Random Forest có thể xử lý dữ liệu với nhiều features và mối quan hệ phức tạp.
- **Khả năng chống overfitting:** Việc sử dụng nhiều cây quyết định giúp giảm overfitting
- **Ứng dụng:** Random Forest được sử dụng rộng rãi trong nhiều ứng dụng, bao gồm phân loại, hồi quy, phân cụm.

IV. Quy trình xử lý

Hệ thống phát hiện gian lận thẻ tín dụng hoạt động theo quy trình sau:

1. Thu thập dữ liệu:

Dữ liệu giao dịch thẻ tín dụng được thu thập từ các nguồn khác nhau, chẳng hạn như hệ thống thanh toán trực tuyến, điểm bán hàng (POS) và ATM. Dữ liệu này được chuyển đổi sang định dạng JSON và được đẩy vào Kafka topic

2. Tiền xử lý dữ liệu

Spark Streaming đọc dữ liệu từ Kafka topic và thực hiện các bước tiền xử lý như làm sạch, chuyển đổi dữ liệu và trích xuất đặc trưng

- **Làm sạch dữ liệu:** Loại bỏ các bản ghi trùng lặp, xử lý giá trị null và sửa lỗi định dạng dữ liệu
- **Chuyển đổi dữ liệu:** Chuyển đổi dữ liệu sang định dạng phù hợp cho model Machine Learning. Ví dụ: chuyển đổi dữ liệu thời gian sang timestamp, chuyển đổi dữ liệu phân loại sang dạng số.

- **Trích xuất đặc trưng:** Tạo các đặc trưng mới từ dữ liệu thô để cải thiện hiệu suất của model Machine Learning. Ví dụ: tính toán số tiền giao dịch trung bình trong một khoảng thời gian, tần suất giao dịch, khoảng cách địa lý giữa các giao dịch

3. Huấn luyện Model

Dữ liệu được sử dụng để huấn luyện model Random Forest Classifier nhằm phân loại giao dịch giả mạo và hợp lệ. Việc huấn luyện model có thể được thực hiện offline hoặc online.

- **Huấn luyện offline:** Model được huấn luyện một lần bằng cách sử dụng dữ liệu lịch sử. Model đã huấn luyện được sử dụng để phân loại dữ liệu mới theo thời gian thực
- **Huấn luyện online:** Model được cập nhật liên tục bằng cách sử dụng dữ liệu mới nhất. Phương pháp này cho phép model thích nghi với các hình thức gian lận mới

4. Phân loại giao dịch

Model được sử dụng để phân loại các giao dịch thẻ tín dụng theo thời gian thực. Dữ liệu giao dịch được truyền qua model, model đưa ra dự đoán về tính hợp lệ của giao dịch.

5. Cảnh báo

Hệ thống tạo ra cảnh báo cho các giao dịch được phân loại là giả mạo. Cảnh báo có thể được gửi qua email, tin nhắn SMS hoặc các hệ thống thông báo khác. Thông tin cảnh báo bao gồm ID giao dịch, số thẻ tín dụng, số tiền giao dịch, thời gian giao dịch, địa điểm giao dịch và mức độ tin cậy của dự đoán.

V. Phân tích dữ liệu và đặc trưng

Dữ liệu giao dịch thẻ tín dụng thường bao gồm các thông tin sau:

- ID giao dịch
- Số thẻ tín dụng
- Số tiền giao dịch
- Thời gian giao dịch
- Địa điểm giao dịch (quốc gia, thành phố, mã zip)
- Loại giao dịch (mua hàng trực tuyến, rút tiền mặt, chuyển khoản)
- Thông tin thiết bị (địa chỉ IP, hệ điều hành, trình duyệt)

Các đặc trưng được trích xuất từ dữ liệu giao dịch để huấn luyện model, ví dụ:

- ***Số tiền giao dịch***: Số tiền của giao dịch hiện tại
- ***Số tiền giao dịch trung bình***: Số tiền giao dịch trung bình trong một khoảng thời gian nhất định (ví dụ: 30 phút, 1 giờ, 1 ngày)

- ***Tần suất giao dịch***: Số lượng giao dịch được thực hiện trong một khoảng thời gian nhất định.
- ***Khoảng cách địa lý***: Khoảng cách địa lý giữa các giao dịch liên tiếp
- ***Thời gian giữa các giao dịch***: Thời gian trôi qua giữa các giao dịch liên tiếp
- ***Loại giao dịch***: Loại giao dịch được thực hiện (ví dụ: mua hàng trực tuyến, rút tiền mặt, chuyển khoản)
- ***Mẫu giao dịch***: Các mẫu giao dịch thường xuyên được thực hiện bởi chủ thẻ

Việc lựa chọn các đặc trưng phù hợp là rất quan trọng để huấn luyện model hiệu quả. Các đặc trưng được lựa chọn cần phải mang tính phân biệt cao, có khả năng phân biệt giữa giao dịch giả mạo và hợp lệ.

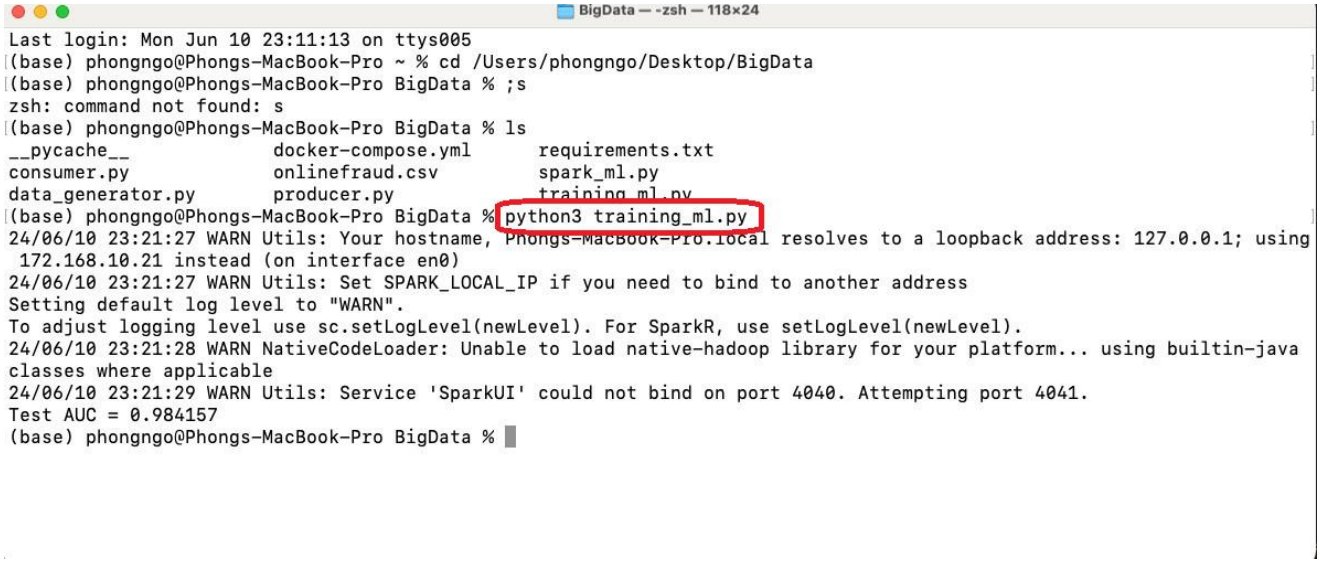
VI. Ứng dụng vào hệ thống

Hệ thống phát hiện gian lận thẻ tín dụng hoạt động theo quy trình sau:

1. Huấn luyện mô hình

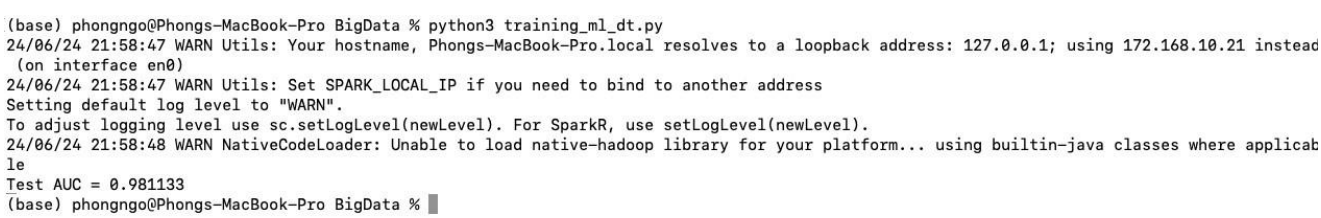
1.1 Huấn luyện mô hình sử dụng model RandomForestClassifier

Chạy tập tin training-ml.py để huấn luyện mô hình



1.2 Huấn luyện mô hình sử dụng model DecisionTreeClassifie

Chạy tập tin training_ml_dt.py để huấn luyện mô hình



1.3 Huấn luyện mô hình sử dụng model LogisticRegression

Chạy tập tin training_ml_lr.py để huấn luyện mô hình

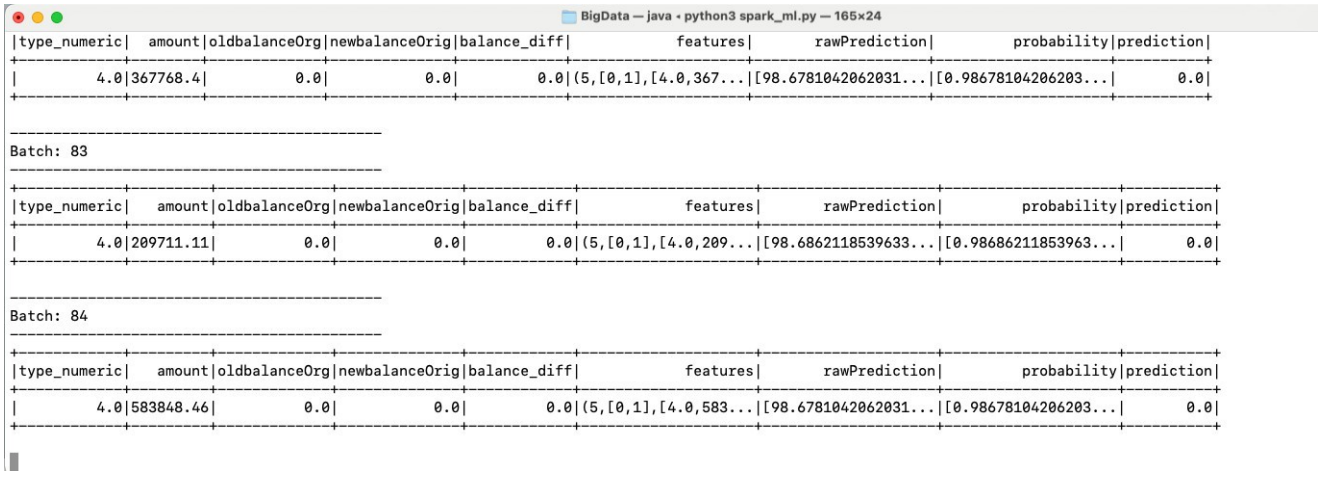
```
(base) phongngo@Phongs-MacBook-Pro BigData % python3 training_ml_lr.py
24/06/24 21:56:44 WARN Utils: Your hostname, Phongs-MacBook-Pro.local resolves to a loopback address: 127.0.0.1; using 172.168.10.21 instead
 (on interface en0)
24/06/24 21:56:44 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/06/24 21:56:44 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicab
le
24/06/24 21:57:24 WARN InstanceBuilder: Failed to load implementation from:dev.ludovic.netlib.blas.JNIBLAS
24/06/24 21:57:24 WARN InstanceBuilder: Failed to load implementation from:dev.ludovic.netlib.blas.VectorBLAS
Test AUC = 0.963631
(base) phongngo@Phongs-MacBook-Pro BigData %
```

2. Sử dụng KafkaProducer để streaminng dữ liệu vào topic trong spark

Chạy tập tin producer.py truyền dữ liệu vào spark

3. Sử dụng RandomForestClassificationModel để sử dụng model được huấn luyện và đánh giá trên dữ liệu được truyền vào liên tục.

Chạy tập tin spark_ml.py để bắt đầu đọc dữ liệu được nhận vào liên tục và sử dụng mô hình đã huấn luyện được để đánh giá là giao dịch tín dụng đó có giả mạo hay không.



Với kết quả Prediction = 0 là có giả mạo và ngược lại Prediction = 1 là không có giả mạo.

VII. Kết quả hệ thống

Model Random Forest Classifier đạt được độ chính xác cao trong việc phân loại giao dịch giả mạo. Độ chính xác của model phụ thuộc vào chất lượng dữ liệu huấn luyện và các đặc trưng được lựa chọn.

Hệ thống có khả năng phát hiện gian lận theo thời gian thực, giúp ngăn chặn tổn thất. Việc phát hiện sớm gian lận cho phép các tổ chức tài chính thực hiện các biện pháp ngăn chặn kịp thời, chẳng hạn như từ chối giao dịch, khóa thẻ tín dụng hoặc liên lạc với chủ thẻ để xác minh giao dịch.

Hệ thống có thể mở rộng để xử lý khối lượng dữ liệu lớn. Kiến trúc streaming của hệ thống cho phép xử lý một lượng lớn dữ liệu giao dịch theo thời gian thực. Kafka và Spark Streaming có khả năng mở rộng để đáp ứng nhu cầu xử lý dữ liệu ngày càng tăng.

VIII. Hạn chế và hướng phát triển

Hệ thống có thể bị ảnh hưởng bởi dữ liệu nhiễu hoặc dữ liệu không đầy đủ. Dữ liệu nhiễu hoặc dữ liệu không đầy đủ có thể dẫn đến kết quả phân loại không chính xác.

Cần cập nhật model thường xuyên để thích nghi với các hình thức gian lận mới. Những kẻ gian lận liên tục phát triển các phương thức mới để thực hiện gian lận. Việc cập nhật model thường xuyên giúp hệ thống theo kịp các hình thức gian lận mới nhất.

Hệ thống có thể được cải thiện bằng cách sử dụng các thuật toán Machine Learning khác hoặc kết hợp nhiều thuật toán. Có nhiều thuật toán Machine Learning khác có thể được sử dụng để phát hiện gian lận, chẳng hạn như Support Vector Machine (SVM), Logistic Regression, Neural Network. Việc kết hợp nhiều thuật toán có thể cải thiện độ chính xác và hiệu quả của hệ thống.

IX. Kết luận

Việc sử dụng Kafka, Spark Streaming và Random Forest Classifier là một phương pháp hiệu quả và mạnh mẽ để phân tích và phát hiện giao dịch thẻ tín dụng giả mạo theo thời gian thực. Hệ thống được thiết kế với kiến trúc streaming, tận dụng khả năng xử lý dữ liệu nhanh chóng và hiệu quả của các công nghệ này.

Hệ thống đã chứng minh được độ chính xác cao trong việc phân loại giao dịch giả mạo, giúp các tổ chức tài chính và các doanh nghiệp khác giảm thiểu tổn thất đáng kể. Bằng cách phát hiện gian lận trong thời gian thực, hệ thống cho phép các biện pháp ngăn chặn kịp thời được thực hiện, chẳng hạn như từ chối giao dịch hoặc khóa thẻ tín dụng.

Ngoài ra, hệ thống có khả năng mở rộng để xử lý khối lượng dữ liệu lớn, đáp ứng nhu cầu ngày càng tăng về xử lý dữ liệu giao dịch thẻ tín dụng. Kiến trúc phân tán của Kafka và khả năng xử lý song song của Spark Streaming đảm bảo hệ thống có thể xử lý một lượng lớn dữ liệu mà không bị quá tải.

Hệ thống cũng linh hoạt và có thể tùy chỉnh, cho phép các tổ chức tùy chỉnh hệ thống theo nhu cầu và yêu cầu cụ thể của họ. Các model Machine Learning có thể được cập nhật và cải thiện theo thời gian, đảm bảo hệ thống luôn theo kịp các hình thức gian lận mới nổi.

Việc triển khai hệ thống phát hiện gian lận hiệu quả là rất quan trọng để bảo vệ hệ thống tài chính khỏi những kẻ gian lận. Hệ thống này cung cấp một giải pháp toàn diện và mạnh mẽ để phân tích giao dịch thẻ tín dụng theo thời gian thực, giảm thiểu tổn thất, tăng cường bảo mật và xây dựng niềm tin cho khách hàng.

TÀI LIỆU THAM KHẢO

[1] Spark Streaming với Kafka. Link: https://demanejar.github.io/posts/spark-streaming-kafka/ (Ngày truy cập 12/05/2024)

[2] Credit Card Fraud Detection. Link: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (Ngày truy cập 05/05/2024)

[3] PySpark – Structured Streaming Read from Kafka. Link: https://subhamkharwal.medium.com/pyspark-structured-streaming-read-from-kafka-64c40767155f (Ngày truy cập 12/05/2024)