Team Name: ByteByByte

Members: Ruchika Suryawanshi, Sumedha Jadhav, Madhura Patil

Theme: Machine Learning Based Fraud Detection System.

—---

Project Title: Real-time Fraud Detection and Transaction Analysis for Transactions

Project Overview:

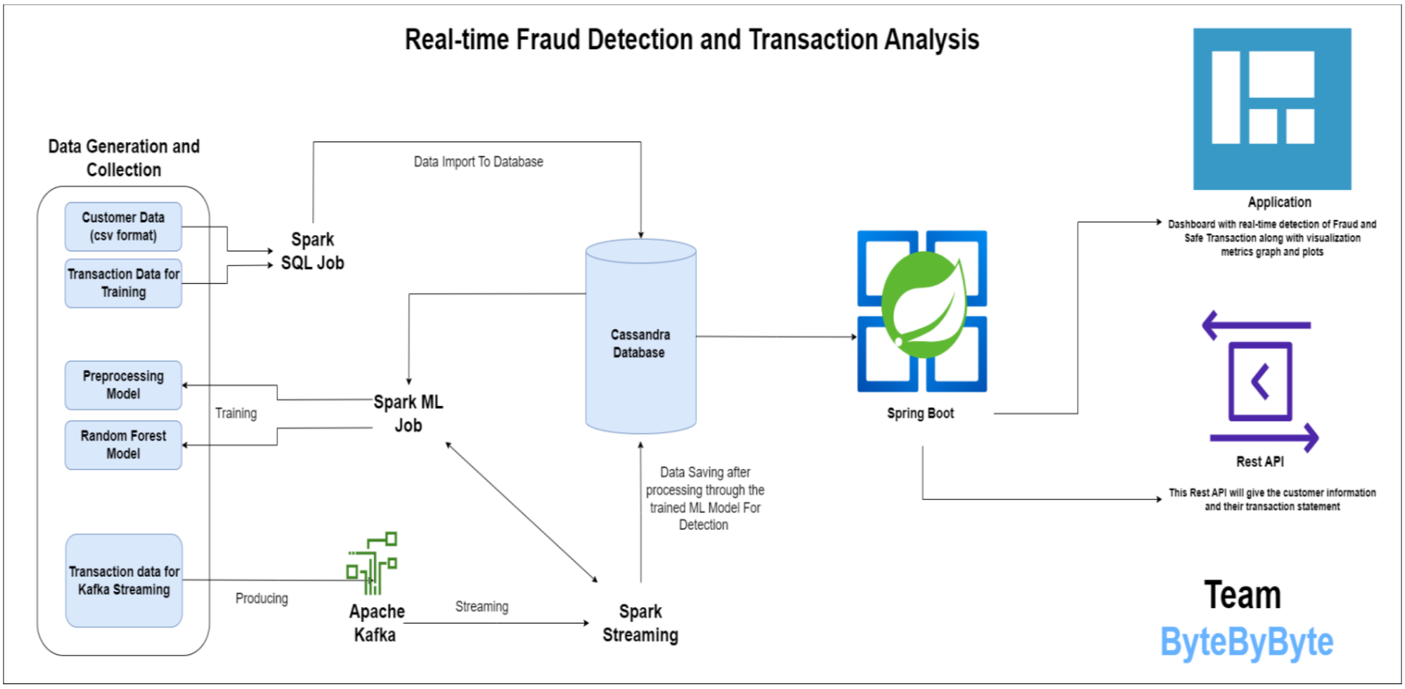
This project creates a real-time fraud detection system that seamlessly combines big data technologies and advanced machine learning. Utilizing Spark, Cassandra, and Spring Boot, the solution accurately spots fraudulent activities, bolstering transaction security and elevating customer experiences. Designed to train on real-time data, it operates alongside live transactions, swiftly identifying and addressing fraud. This approach aligns with the "Machine Learning Based Fraud Detection System" theme, harnessing machine learning, real-time data processing, and streamlined visualization for proactive fraud prevention, enhancing financial security and transaction integrity.

Proposed Solution Highlights Include:

1. **Streamlined Data Processing:** The project seamlessly processes customer data and transaction records stored in CSV files to enable real-time fraud detection.
2. **Machine Learning Techniques:** By harnessing machine learning methodologies, the system is equipped to accurately identify and classify fraudulent transactions with a high degree of accuracy.
3. **Kafka-powered Real-time Processing:** To ensure responsiveness, the system leverages Kafka for streamlined data streaming, allowing it to efficiently handle incoming transactions as they occur.
4. **Immediate Fraud Response:** This capability enables prompt classification of new transactions, facilitating an instantaneous response to potential fraudulent activities.
5. **User-Friendly Dashboard:** The system incorporates a user-friendly dashboard for visualizing classified transactions in real-time, empowering credit card companies to take swift and informed actions.
6. **Enhanced User Experience:** Convenient REST APIs simplify customer information retrieval and the generation of transaction statements, elevating the overall user experience and system functionality.

High-level Architecture:

The high-level architecture of the "Machine Learning Based Fraud Detection System" project encompasses a series of interconnected components and processes that work together to achieve efficient real-time fraud detection. The architecture leverages cutting-edge technologies to handle data generation, preprocessing, machine learning model training, streaming, visualization, and interaction through APIs. Here is an overview of the architecture:


Real-time Fraud Detection and Transaction Analysis
Data Generation and Collection
Customer Data (csv format)
Transaction Data for Training
Preprocessing Model
Random Forest Model
Transaction data for Kafka Streaming
Spark SQL Job
Data Import To Database
Cassandra Database
Spark ML Job
Training
Producing
Apache Kafka
Streaming
Spark Streaming
Data Saving after processing through the trained ML Model For Detection
Spring Boot
Application
Dashboard with real-time detection of Fraud and Safe Transaction along with visualization metrics graph and plots
Rest API
This Rest API will give the customer information and their transaction statement
Team
ByteByByte

1. Data Generation & Storage:

- Simulated customer data and transactions are generated and stored in CSV files.
- Spark SQL imports data into Cassandra, organized in tables.

2. Machine Learning Model:

- Spark ML preprocesses data and trains a fraud detection model using the Random Forest algorithm
- The trained model is saved in the filesystem.

3. Real-time Streaming:

- Kafka streams transaction data using topics.

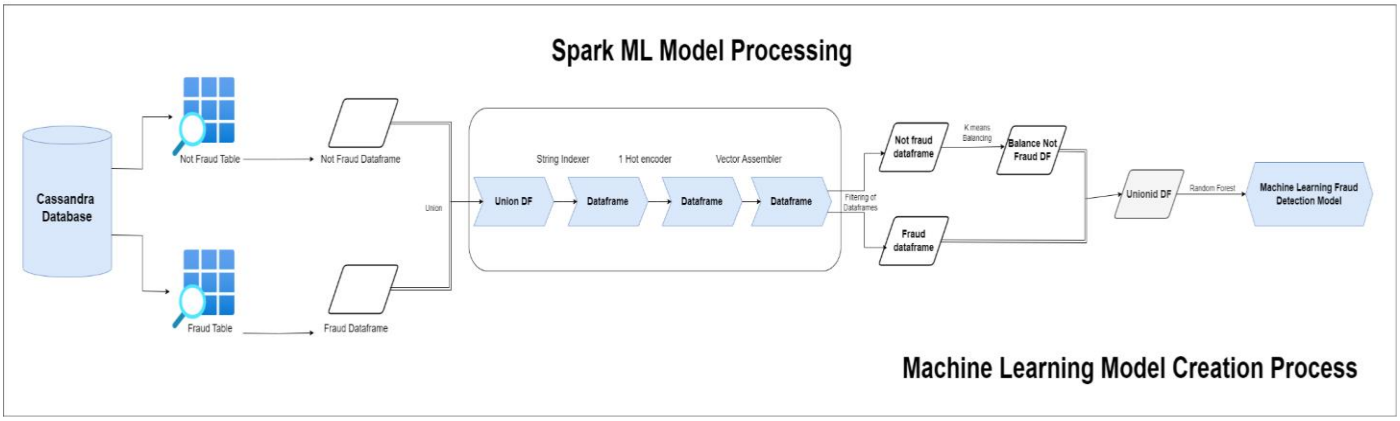
4. Spark Streaming:

- Spark Streaming processes Kafka data. The pre-trained model predicts fraud, saving results in Cassandra.

5. Dashboard & APIs:

- Spring Boot creates a dashboard for real-time fraud insights.
- Spring Boot based Rest APIs allow interaction for data retrieval and transaction statements.

Machine Learning Implementation:



1. Data Retrieval & Preprocessing:

- Extract customer and transaction data using Spark SQL.
- Import data to Cassandra, computing age and distance.

2. Data Split & Storage:

- Separate fraud and non-fraud data, stored in Cassandra.

3. Data Fusion & Transformation:

- Load and unite fraud/non-fraud data frames.
- Apply Spark ML Pipeline for transformations.

4. Feature Transformation:

- Convert columns using String Indexer to double values.
- Normalize values with One Hot Encoder.

5. Feature Assembly:

- Assemble features into a vector using Vector Assembler.

6. Algorithm Training & Balancing:

- Train algorithm on assembled data.
- Balance data using K-means for accurate training.

7. Data Balancing & Model Creation:

- Combine balanced data frames.
- Apply Random Forest for classification.

8. Model Saving:

- Save a model to utilize it for fraud detection from the Kafka stream Transaction

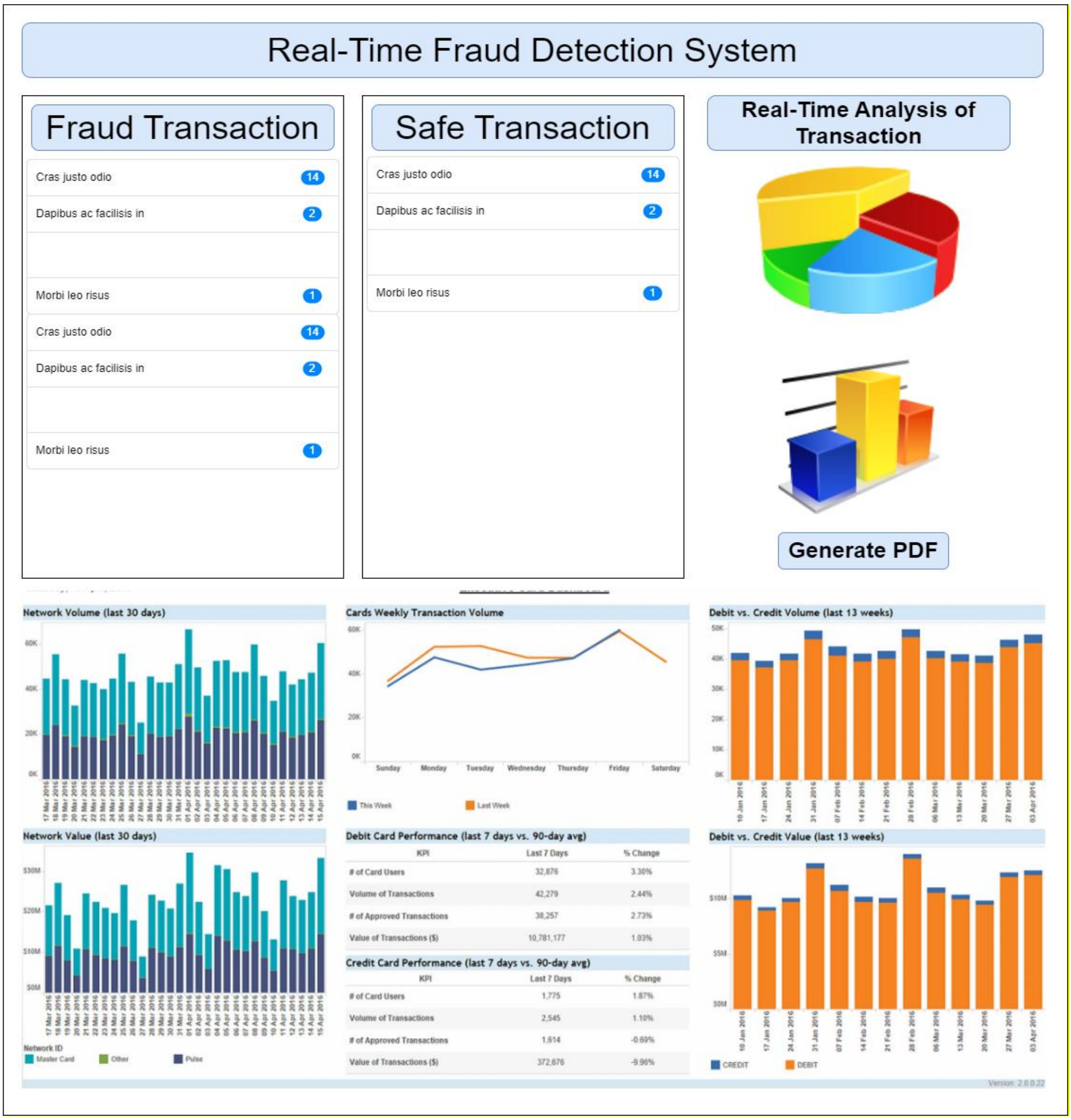
Apache Kafka Streaming Transaction Implementation:

1. The Spark Streaming job consumes transaction messages from Kafka's "Transaction" topic.
2. For each message, it calculates customer age and merchant-customer distance from Cassandra data, using them as prediction features.
3. Preprocessing and Random Forest models are loaded to predict fraudulence.
4. Predicted transactions are saved in **Cassandra's** "fraud" or "non-fraud" tables.
5. Kafka offsets ensure reliable, "exactly once" semantics.

Real Time Application Interface Implementation:

1. **Spring Boot-based** dashboard fetches and displays real-time data from Cassandra.
2. Latest fraud and non-fraud transactions within 5 seconds are queried.
3. Max timestamp maintains data coherence.
4. Only transactions exceeding the previous max timestamp are displayed.
5. Users receive up-to-date insights into fraud and non-fraud transactions.

<u>This is the rough overview of the application which we will be implementing for the Hackathon (Created using Draw.io Prototyping Tool)</u>



Debit Card Performance (last 7 days vs. 90-day avg)

KPI	Last 7 Days	% Change
# of Card Users	32,876	3.30%
Volume of Transactions	42,279	2.44%
# of Approved Transactions	38,257	2.73%
Value of Transactions ($)	10,781,177	1.83%

Credit Card Performance (last 7 days vs. 90-day avg)

KPI	Last 7 Days	% Change
# of Card Users	1,775	1.87%
Volume of Transactions	2,545	1.10%
# of Approved Transactions	1,614	-0.69%
Value of Transactions ($)	372,676	-9.96%