|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| Fraud Transaction Prediction |
| |  |  |  | | --- | --- | --- | | **SPA ASSIGNMENT GROUP 43** |  | Stream Processing & Analytics | |

Contents

[**Introduction** 3](#_Toc43328840)

[**Work Preparation \ Data Flow Diagrams** 3](#_Toc43328841)

[**Kafka Setup:** 4](#_Toc43328842)

[**Kafka Topic Creation** 4](#_Toc43328843)

[**Kafka Topic List** 5](#_Toc43328844)

[**Data Exploration:** 5](#_Toc43328845)

[**Pre-Processing:** 6](#_Toc43328846)

[**List of Spark Programs Developed** 6](#_Toc43328847)

[**Spark Programs Explained Step by Step** 7](#_Toc43328848)

[**PublishToKafkaTopic:** 7](#_Toc43328849)

[**What It Does?** 7](#_Toc43328850)

[**How It Does?** 7](#_Toc43328851)

[**Input Parameters to program:** 7](#_Toc43328852)

[**Code Snippets:** 7](#_Toc43328853)

[**SubscribeToKafkaTopic:** 7](#_Toc43328854)

[**What it does?** 7](#_Toc43328855)

[**How It Does?** 7](#_Toc43328856)

[**Input Params to program:** 7](#_Toc43328857)

[**Code Snippets:** 8](#_Toc43328858)

[**LogRegSparkMLPipeline:** 8](#_Toc43328859)

[**What it does?** 8](#_Toc43328860)

[**How it does?** 8](#_Toc43328861)

[**Input Params to Program:** 8](#_Toc43328862)

[**Code Snippets:** 8](#_Toc43328863)

[**Code Snippet to Evaluate Model and display Accuracy:** 9](#_Toc43328864)

[**Code Snippet to save model path:** 9](#_Toc43328865)

[**PredictRealTime** 9](#_Toc43328866)

[**What it Does?** 9](#_Toc43328867)

[**How it Does?** 9](#_Toc43328868)

[**Input Params:** 10](#_Toc43328869)

[**Code Snippets:** 10](#_Toc43328870)

[**Code Snippet to subscribe to Kafka topic:** 10](#_Toc43328871)

[**Code Snippet to Load ML Model Saved in previous step:** 10](#_Toc43328872)

[**Code Snippet to save transaction to different Kafka topics based on Prediction** 10](#_Toc43328873)

[**Output** 11](#_Toc43328874)

[**Fraud Transactions Observed in Kafka Topic, “FraudTxns”:** 11](#_Toc43328875)

[**Normal Transactions captured into Kafka Topic “NormalTxns”:** 11](#_Toc43328876)

# **Introduction**

Problem Statement: **Predicting Fraud Transactions**

Tech Stack: **Kafka, Spark**

Programming APIs: **Spark Dataframes, Structured Streaming, MLLib, Kafka APIs**

Programming Language: **Scala**

Project Build Tool: **Maven**

IDE used for Development: **IntelliJ**

Git Location of Code: <https://github.com/mighty-raj/SPA_Streaming_Assignment.git>

Training Data Used: <https://www.kaggle.com/ntnu-testimon/paysim1>

Team Name: **SPA ASSIGNMENT GROUP 43**

Team Members:

Mrutyamjaya Surampudi,

Srinivas Veerabomma,

Raja Mahesh Aravapalli

# **Work Preparation \ Data Flow Diagrams**

A close up of text on a whiteboard

Description automatically generated

A close up of text on a whiteboard

Description automatically generated

A close up of text on a whiteboard

Description automatically generated

A close up of text on a whiteboard

Description automatically generated

# **Kafka Setup:**

## **Kafka Topic Creation**

kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic **masterData**

kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic **testDataTopic**

kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic **FraudTxns**

kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic **NormalTxns**

## **Kafka Topic List**

A picture containing bird

Description automatically generated

# **Data Exploration:**

Used “Spylon-Kernal” in JupyterA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A close up of a logo

Description automatically generated

A screenshot of a social media post

Description automatically generated

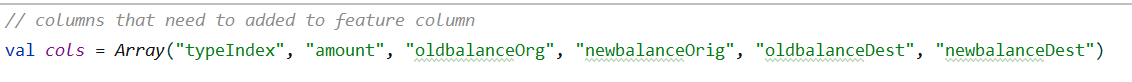
A screenshot of a cell phone

Description automatically generated

# **Pre-Processing:**

Based on the observations from Data Exploration, performed only below transformations on dataset before applying Logistic Regression Machine Learning Model:

1. Considered Below columns for modelling



1. As per the dataset given, it is observed, “**isFraud**” is flagged only for Transaction types, TRANSFER & CASH\_OUT. Hence, filtered-out other types of transactions from model training.  
   A screenshot of a cell phone

   Description automatically generated
2. Since column “type” is observed to be a Categorical value, transformed it to numerical for to pass for model training…. Using **StringIndexer** in spark mllib.  
   A screenshot of a cell phone

   Description automatically generated
3. Done few more as required for to apply ML modelling …

# **List of Spark Programs Developed**

As part of this project we created 4 Spark Programs and given below are the list:

1. PublishToKafkaTopic
2. SubscribeToKafkaTopic
3. LogRegSparkMLPipeline
4. PredictRealTime

# **Spark Programs Explained Step by Step**

## **PublishToKafkaTopic:**

### **What It Does?**

Reads a CSV source files from source directory path as it streams and writes content to Kafka Topic in JSON Format in real time.

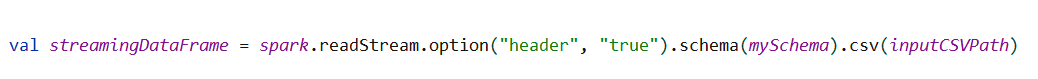
### **How It Does?**

Using Spark Structured Streaming API calls, application publishes data to Kafka topic from source directory as the new files arrives at source location.

### **Input Parameters to program:**

1. Path to the CSV Source File
2. Kafka Broker Host/IP & Port
3. Kafka Topic Name to write data to.
4. Checkpoint directory path, to recover from Failures

### **Code Snippets:**



A screenshot of a cell phone

Description automatically generated

## **SubscribeToKafkaTopic:**

### **What it does?**

Reads Kafka Topic and streams data into a Destination Path (Master Data) given.

### **How It Does?**

Using Spark Structured streaming APIs, subscribes to a kafka topic and writes data to given destination path.

### **Input Params to program:**

1. Kafka Topic Name
2. Directory location to stream data into.
3. Checkpoint directory path, to recover from Failures

### **Code Snippets:**

A screenshot of a cell phone

Description automatically generated

## **LogRegSparkMLPipeline:**

### **What it does?**

1. Trains a machine learning model that can Predict fraud transactions.
2. Uses PaySim data to train the model
3. Applies Logistic Regression ml technique.

### **How it does?**

Using Spark ML Lib Pipeline APIs, trains a logistic regression ml model, and saves trained model to given path, which can later be used to predict on unseen transaction!

### **Input Params to Program:**

1. Input Directory Path for training data
2. Directory path to save the trained model

### **Code Snippets:**

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

### **Code Snippet to Evaluate Model and display Accuracy:**

A screenshot of a cell phone

Description automatically generated

### **Code Snippet to save model path:**

A picture containing knife

Description automatically generated

## **PredictRealTime**

### **What it Does?**

1. Loads saved ML Model
2. Reads messages in real time from kafka topic
3. Applies ML Model and predicts transaction in real-time
4. Depending the prediction, weather it is a Fraud or Normal transaction, writes back the data to different kafka topics configured.

### **How it Does?**

Using Spark MLLib and Spark Structured streaming APIs, load ml model from saved directory, and then applies it to incoming kafka messages arriving into kafka in real-time

### **Input Params:**

1. Input Directory Path to ML Model to load from
2. Source Kafka topic to read incoming transactions in real-time
3. Kafka Topic Name to write Fraud Transactions
4. Kafka Topic Name to write Normal Transactions

### **Code Snippets:**

### **Code Snippet to subscribe to Kafka topic:**

A screenshot of a cell phone

Description automatically generated

### **Code Snippet to Load ML Model Saved in previous step:**

A screenshot of a cell phone

Description automatically generated

### **Code Snippet to save transaction to different Kafka topics based on Prediction**

A screenshot of a cell phone

Description automatically generated

# **Output**

## **Fraud Transactions Observed in Kafka Topic, “FraudTxns”:**

A picture containing bottle, keyboard, computer

Description automatically generated

## **Normal Transactions captured into Kafka Topic “NormalTxns”:**

A blue and white text

Description automatically generated