# Final Project

Big Data System & Intelligence Analytics

# Serverless Transactions Fraud Detection & Notification

**December 18, 2020**

Team-2

- Manish Singh
- Phu Tran
- Kaviya Sachi
- Rajavi Mehta

## Overview

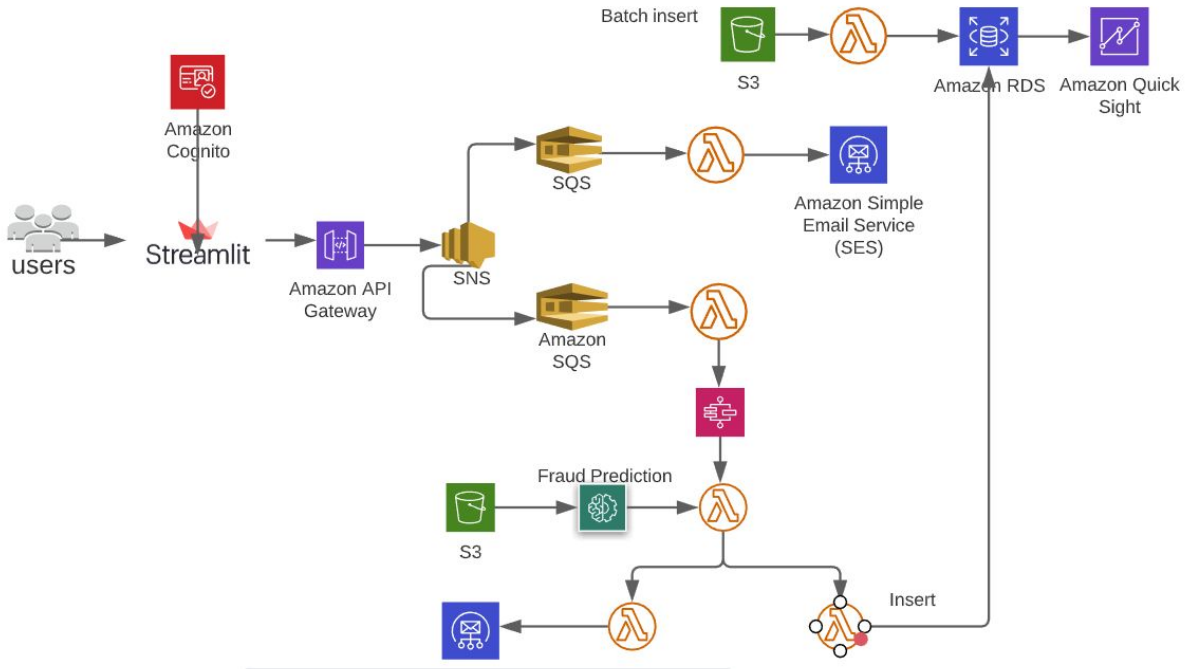
Did you know that each year, tens of billions of dollars are lost to online fraud world-wide?

Companies with online businesses have to constantly be on guard for fraudulent activity such as fake accounts and payments made with stolen credit cards. One way they try to identify fraudsters is by using fraud detection apps, some of which use Machine Learning (ML).

Transaction Fraud Detection is a fully managed service that makes it easy to identify potentially fraudulent online Transaction such as online payment fraud.

Enter our Transaction Fraud Detection app. It uses historical data, ML to identify potentially fraudulent online activity so you can catch more fraud faster. You can create a fraud detection model with just a few clicks and no prior ML experience because Fraud Detector handles all of the ML heavy lifting for you.

# Architecture



**Architecture components:**

- Streamlit app
- Amazon cognito
- Api Gateway
- Amazon SNS
- Amazon SQS
- Amazon SES
- Step Function
- Lambda Function
- Amazon Fraud Detector
- Amazon S3
- Amazon RDS
- Amazon Quick Sight

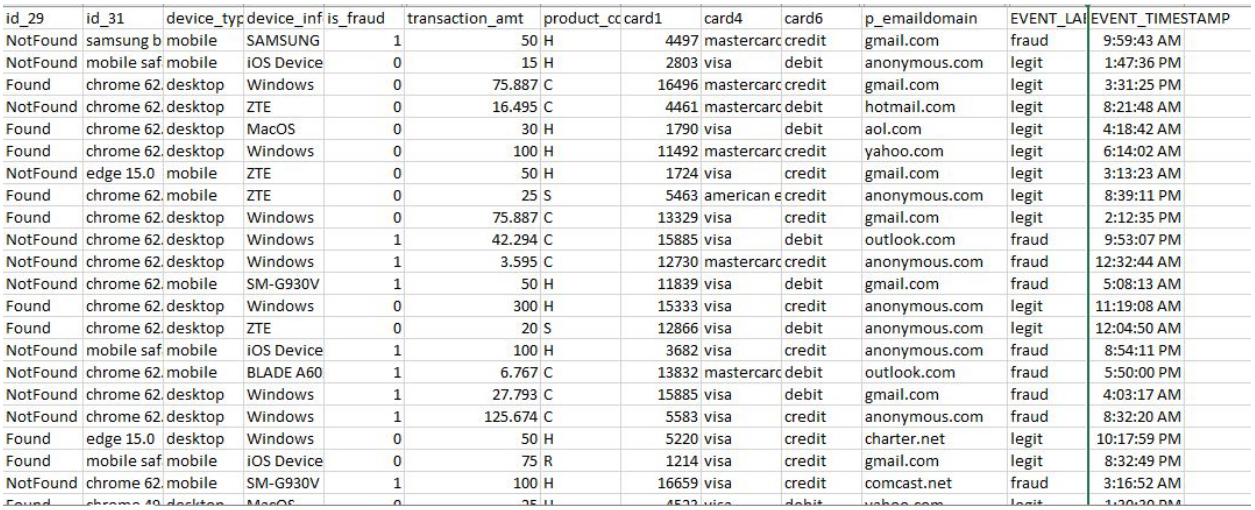
**SQS** is mainly used to decouple applications or integrate applications. Messages can be stored in **SQS** for short duration of time (max 14 days). **SNS** distributes several copies of message to several subscribers

# Dataset

Link : https://www.kaggle.com/c/ieee-fraud-detection/data

There are 19 columns in the dataset:

Transaction_id - transaction ID
Id_29 - if the transaction was found or not
Id_31 - browser type(Chrome, windows, safari)
Device_type - Type of device(Mobile, Desktop)
Device_info - information about the device(MacOS, Android, Windows, etc.)
is_fraud(Target variable)
Transaction_dt
Transaction_amt - amount in the transaction
Product_cd
Card1 - card number
Card4 - card type(Mastercard, visa)
Card6 - card type(Debit, credit)
Addr1 - address 1
Addr2 - address 2
P_emaildomain - domain of email_id
Time - Time of transaction
EVENT_LABEL - describes if the event was fraud or legit
Date - date of transaction
EVENT_TIMESTAMP - timestamp of event



| id_29 | id_31 | device_typ | device_inf | is_fraud | transaction_amt | product_cc | card1 | card4 | card6 | p_emaildomain | EVENT_LAI | EVENT_TIMESTAMP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NotFound | samsung b | mobile | SAMSUNG | 1 | 50 | H | 4497 | mastercarc | credit | gmail.com | fraud | 9:59:43 AM |
| NotFound | mobile saf | mobile | iOS Device | 0 | 15 | H | 2803 | visa | debit | anonymous.com | legit | 1:47:36 PM |
| Found | chrome 62. | desktop | Windows | 0 | 75.887 | C | 16496 | mastercarc | credit | gmail.com | legit | 3:31:25 PM |
| NotFound | chrome 62. | desktop | ZTE | 0 | 16.495 | C | 4461 | mastercarc | debit | hotmail.com | legit | 8:21:48 AM |
| Found | chrome 62. | desktop | MacOS | 0 | 30 | H | 1790 | visa | debit | aol.com | legit | 4:18:42 AM |
| Found | chrome 62. | desktop | Windows | 0 | 100 | H | 11492 | mastercarc | credit | yahoo.com | legit | 6:14:02 AM |
| NotFound | edge 15.0 | mobile | ZTE | 0 | 50 | H | 1724 | visa | credit | gmail.com | legit | 3:13:23 AM |
| Found | chrome 62. | mobile | ZTE | 0 | 25 | S | 5463 | american e | credit | anonymous.com | legit | 8:39:11 PM |
| Found | chrome 62. | desktop | Windows | 0 | 75.887 | C | 13329 | visa | credit | gmail.com | legit | 2:12:35 PM |
| NotFound | chrome 62. | desktop | Windows | 1 | 42.294 | C | 15885 | visa | debit | outlook.com | fraud | 9:53:07 PM |
| NotFound | chrome 62. | desktop | Windows | 1 | 3.595 | C | 12730 | mastercarc | credit | anonymous.com | fraud | 12:32:44 AM |
| NotFound | chrome 62. | mobile | SM-G930V | 1 | 50 | H | 11839 | visa | debit | gmail.com | fraud | 5:08:13 AM |
| Found | chrome 62. | desktop | Windows | 0 | 300 | H | 15333 | visa | credit | anonymous.com | legit | 11:19:08 AM |
| Found | chrome 62. | desktop | ZTE | 0 | 20 | S | 12866 | visa | debit | anonymous.com | legit | 12:04:50 AM |
| NotFound | mobile saf | mobile | iOS Device | 1 | 100 | H | 3682 | visa | credit | anonymous.com | fraud | 8:54:11 PM |
| NotFound | chrome 62. | mobile | BLADE A60 | 1 | 6.767 | C | 13832 | mastercarc | debit | outlook.com | fraud | 5:50:00 PM |
| NotFound | chrome 62. | desktop | Windows | 1 | 27.793 | C | 15885 | visa | debit | gmail.com | fraud | 4:03:17 AM |
| NotFound | chrome 62. | desktop | Windows | 1 | 125.674 | C | 5583 | visa | credit | anonymous.com | fraud | 8:32:20 AM |
| Found | edge 15.0 | desktop | Windows | 0 | 50 | H | 5220 | visa | credit | charter.net | legit | 10:17:59 PM |
| Found | mobile saf | mobile | iOS Device | 0 | 75 | R | 1214 | visa | credit | gmail.com | legit | 8:32:49 PM |
| NotFound | chrome 62. | mobile | SM-G930V | 1 | 100 | H | 16659 | visa | credit | comcast.net | fraud | 3:16:52 AM |
| Found | chrome 49 | desktop | MacOS | 0 | 25 | H | 4523 | visa | debit | yahoo.com | legit | 1:20:20 PM |

# Use cases

- Batch Process : Ingest data from S3 to RDS via lambda function and connect to quicksight for data visualization
- The user signups via streamlit. The streamlit is connected to cognito which helps in verifying user and generating password
- When user logins, token is generated, then he/she is allowed inside the application
- The apigateway helps to create endpoint
- SNS helps with the parallel processing
- SQS helps with the queueing
- The lambda function connects to email service, for sending order confirmation
- The step function helps to parallel processing of Fraud detection and insertion of records in RDS
- And finally an email confirmation to user

# Making the Fraud Detector Model

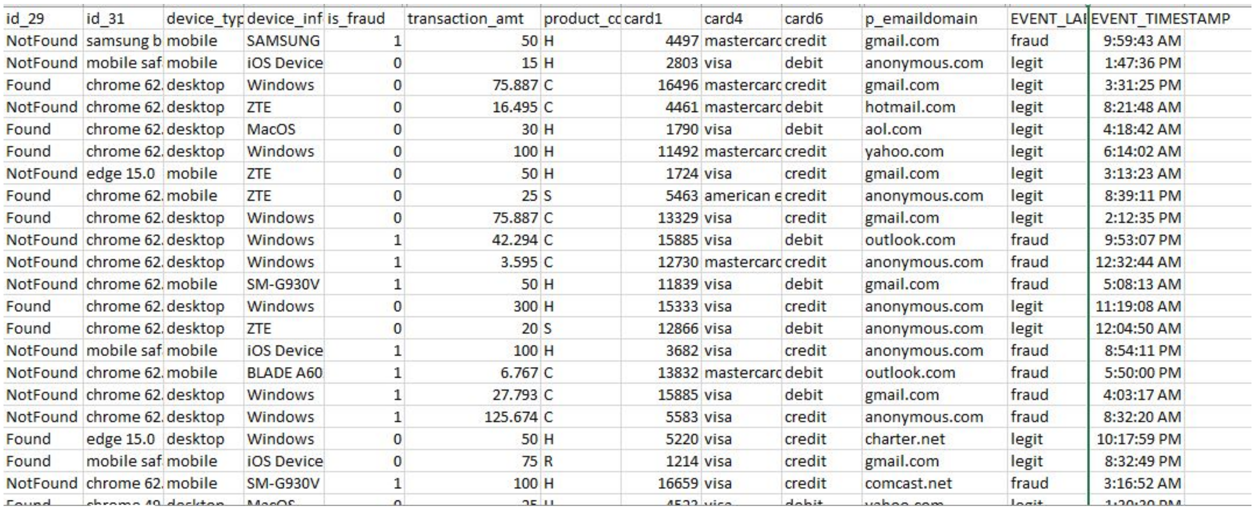
Step 1: Define the event you want to assess for fraud.

Amazon Fraud Detector is a fully managed service that makes it easy to identify potentially fraudulent online activities such as online payment fraud and the creation of fake accounts.

Pay only for what you use. There are no minimum

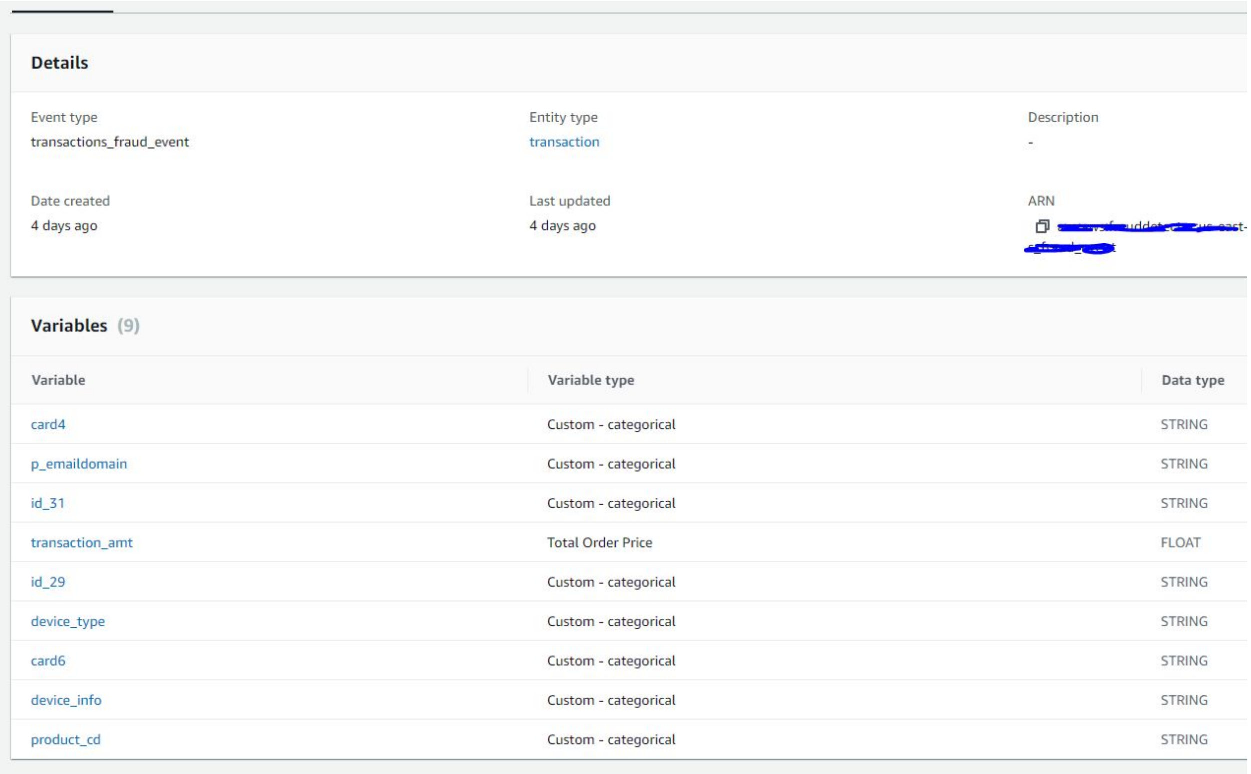
How it works

Step 2: Upload your historical event dataset to Amazon S3 and select a fraud detection model type.



| id_29 | id_31 | device_typ | device_inf | is_fraud | transaction_amt | product_cc | card1 | card4 | card6 | p_emaildomain | EVENT_LAI | EVENT_TIMESTAMP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NotFound | samsung b | mobile | SAMSUNG | 1 | 50 | H | 4497 | mastercard | credit | gmail.com | fraud | 9:59:43 AM |
| NotFound | mobile saf | mobile | iOS Device | 0 | 15 | H | 2803 | visa | debit | anonymous.com | legit | 1:47:36 PM |
| Found | chrome 62. | desktop | Windows | 0 | 75.887 | C | 16496 | mastercard | credit | gmail.com | legit | 3:31:25 PM |
| NotFound | chrome 62. | desktop | ZTE | 0 | 16.495 | C | 4461 | mastercard | debit | hotmail.com | legit | 8:21:48 AM |
| Found | chrome 62. | desktop | MacOS | 0 | 30 | H | 1790 | visa | debit | aol.com | legit | 4:18:42 AM |
| Found | chrome 62. | desktop | Windows | 0 | 100 | H | 11492 | mastercard | credit | yahoo.com | legit | 6:14:02 AM |
| NotFound | edge 15.0 | mobile | ZTE | 0 | 50 | H | 1724 | visa | credit | gmail.com | legit | 3:13:23 AM |
| Found | chrome 62. | mobile | ZTE | 0 | 25 | S | 5463 | american e | credit | anonymous.com | legit | 8:39:11 PM |
| Found | chrome 62. | desktop | Windows | 0 | 75.887 | C | 13329 | visa | credit | gmail.com | legit | 2:12:35 PM |
| NotFound | chrome 62. | desktop | Windows | 1 | 42.294 | C | 15885 | visa | debit | outlook.com | fraud | 9:53:07 PM |
| NotFound | chrome 62. | desktop | Windows | 1 | 3.595 | C | 12730 | mastercard | credit | anonymous.com | fraud | 12:32:44 AM |
| NotFound | chrome 62. | mobile | SM-G930V | 1 | 50 | H | 11839 | visa | debit | gmail.com | fraud | 5:08:13 AM |
| Found | chrome 62. | desktop | Windows | 0 | 300 | H | 15333 | visa | credit | anonymous.com | legit | 11:19:08 AM |
| Found | chrome 62. | desktop | ZTE | 0 | 20 | S | 12866 | visa | debit | anonymous.com | legit | 12:04:50 AM |
| NotFound | mobile saf | mobile | iOS Device | 1 | 100 | H | 3682 | visa | credit | anonymous.com | fraud | 8:54:11 PM |
| NotFound | chrome 62. | mobile | BLADE A60 | 1 | 6.767 | C | 13832 | mastercard | debit | outlook.com | fraud | 5:50:00 PM |
| NotFound | chrome 62. | desktop | Windows | 1 | 27.793 | C | 15885 | visa | debit | gmail.com | fraud | 4:03:17 AM |
| NotFound | chrome 62. | desktop | Windows | 1 | 125.674 | C | 5583 | visa | credit | anonymous.com | fraud | 8:32:20 AM |
| Found | edge 15.0 | desktop | Windows | 0 | 50 | H | 5220 | visa | credit | charter.net | legit | 10:17:59 PM |
| Found | mobile saf | mobile | iOS Device | 0 | 75 | R | 1214 | visa | credit | gmail.com | legit | 8:32:49 PM |
| NotFound | chrome 62. | mobile | SM-G930V | 1 | 100 | H | 16659 | visa | credit | comcast.net | fraud | 3:16:52 AM |
| Found | chrome 49. | desktop | MacOS | 0 | 25 | H | 4522 | visa | debit | yahoo.com | legit | 1:30:30 PM |

Step 3: Amazon Fraud Detector uses your historical data as input to build a custom model. The service automatically inspects and enriches data, performs feature engineering, selects algorithms, trains and tunes your model, and hosts the model.



**Details**

Event type
transactions_fraud_event

Entity type
transaction

Description
-

Date created
4 days ago

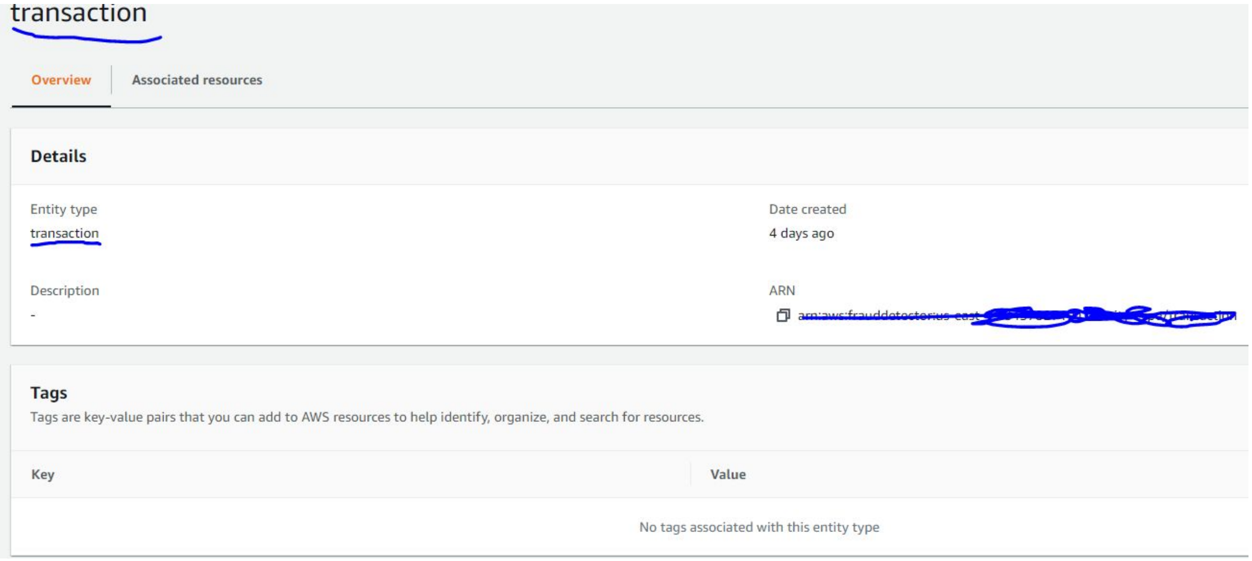
Last updated
4 days ago

ARN
[redacted]

**Variables** (9)

| Variable | Variable type | Data type |
|---|---|---|
| card4 | Custom - categorical | STRING |
| p_emaildomain | Custom - categorical | STRING |
| id_31 | Custom - categorical | STRING |
| transaction_amt | Total Order Price | FLOAT |
| id_29 | Custom - categorical | STRING |
| device_type | Custom - categorical | STRING |
| card6 | Custom - categorical | STRING |
| device_info | Custom - categorical | STRING |
| product_cd | Custom - categorical | STRING |

Create entity:



transaction

**Details**

Entity type
transaction

Date created
4 days ago

Description
-

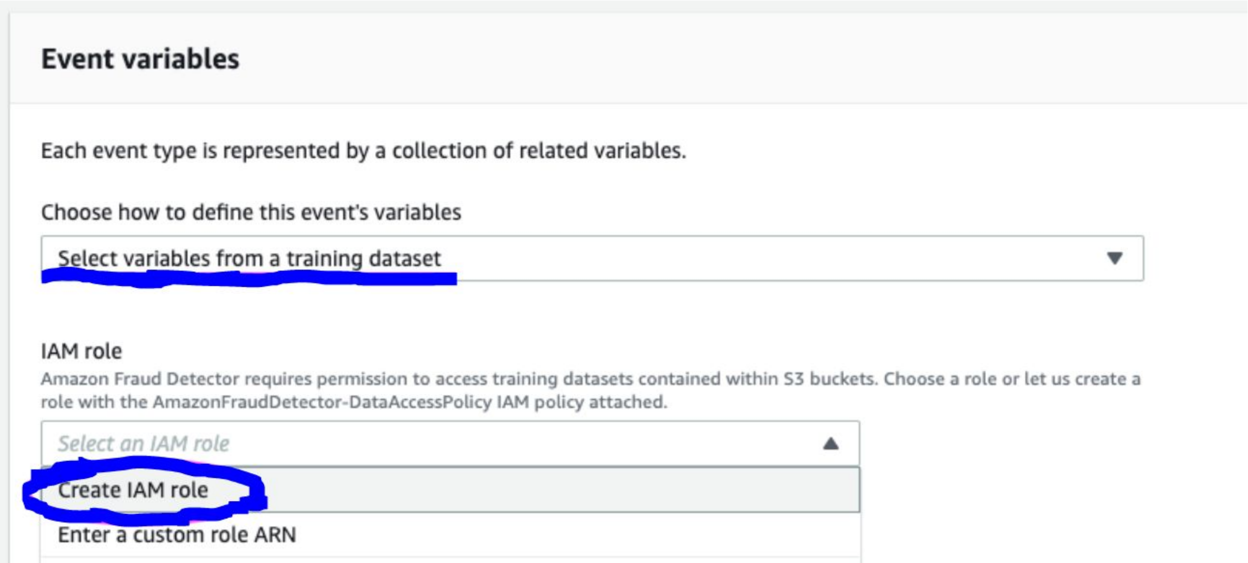
ARN
[redacted]

**Tags**

Tags are key-value pairs that you can add to AWS resources to help identify, organize, and search for resources.

| Key | Value |
|---|---|
| No tags associated with this entity type | |

We move on to Event Variables. We will select variables from a training dataset. This will allow us to use the earlier mentioned CSV file and pull in the headers.



Event variables

Each event type is represented by a collection of related variables.

Choose how to define this event's variables

Select variables from a training dataset ▼

IAM role

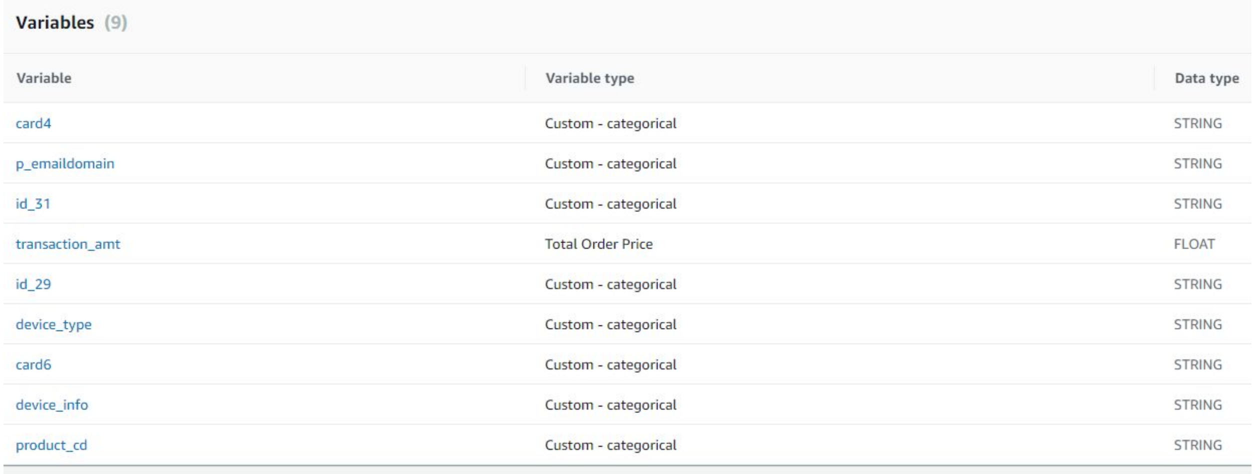
Amazon Fraud Detector requires permission to access training datasets contained within S3 buckets. Choose a role or let us create a role with the AmazonFraudDetector-DataAccessPolicy IAM policy attached.

Select an IAM role ▲

[✓ Create IAM role]

Enter a custom role ARN

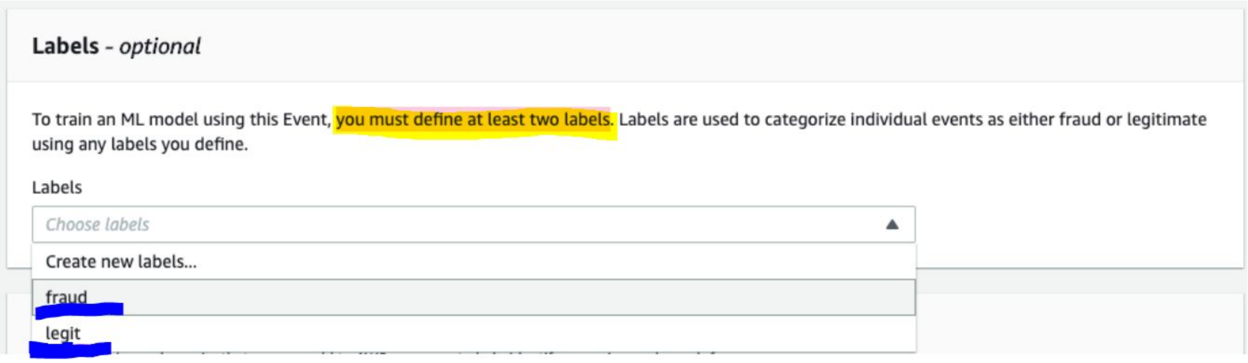
And now we can upload the earlier mentioned CSV file to pull in the headers



Variables (9)

| Variable | Variable type | Data type |
|---|---|---|
| card4 | Custom - categorical | STRING |
| p_emaildomain | Custom - categorical | STRING |
| id_31 | Custom - categorical | STRING |
| transaction_amt | Total Order Price | FLOAT |
| id_29 | Custom - categorical | STRING |
| device_type | Custom - categorical | STRING |
| card6 | Custom - categorical | STRING |
| device_info | Custom - categorical | STRING |
| product_cd | Custom - categorical | STRING |

Because we are going to define a model, we must *define at least two labels*



**Labels** - *optional*

To train an ML model using this Event, you must define at least two labels. Labels are used to categorize individual events as either fraud or legitimate using any labels you define.

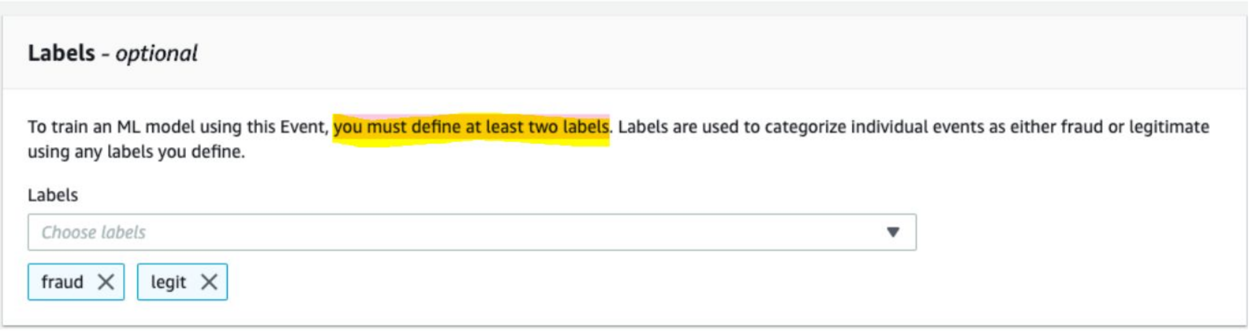
Labels

*Choose labels* ▲

Create new labels...

fraud

legit



**Labels** - *optional*

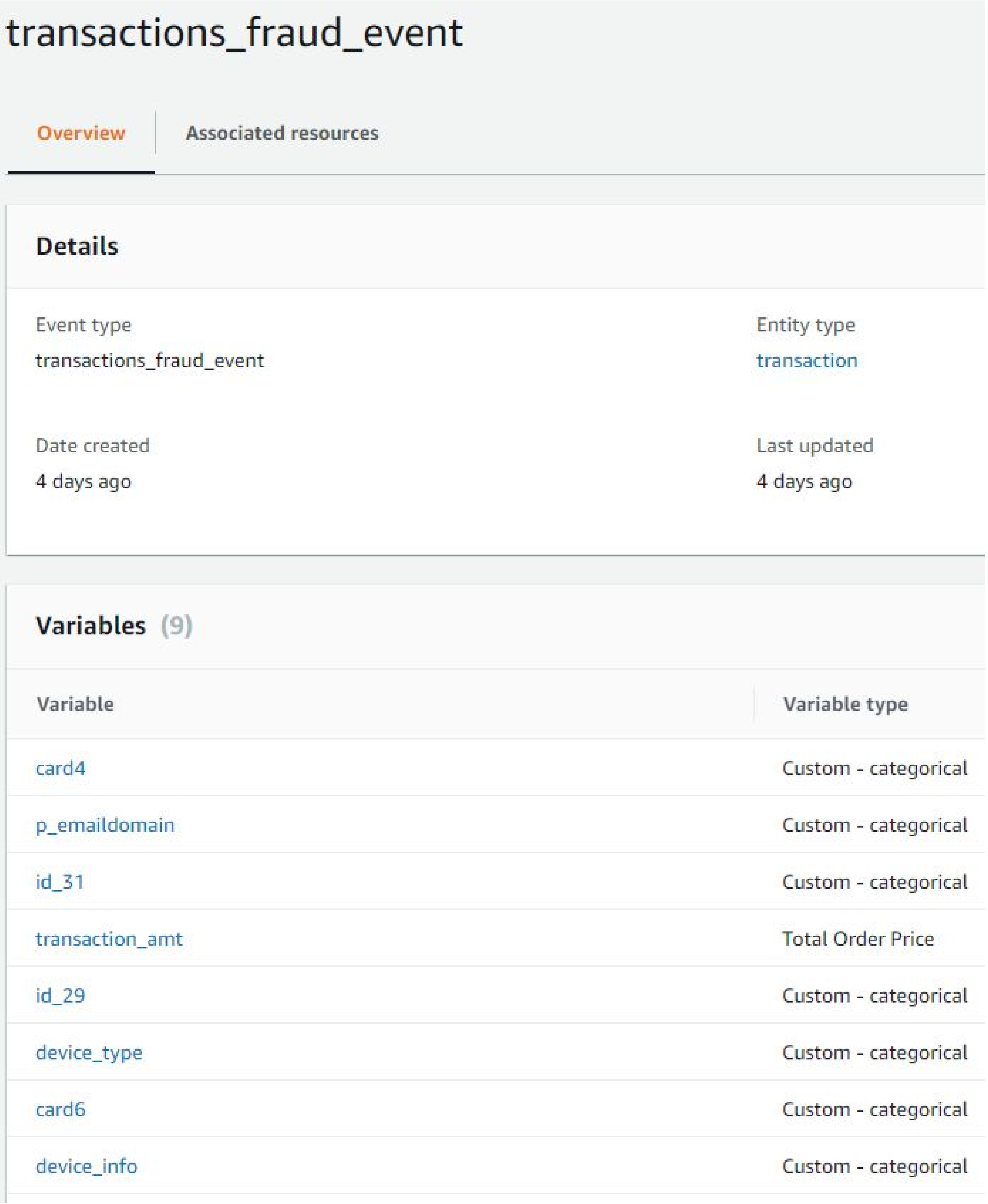
To train an ML model using this Event, you must define at least two labels. Labels are used to categorize individual events as either fraud or legitimate using any labels you define.

Labels

*Choose labels* ▼

fraud ✕ legit ✕

If all goes well, we get a *happy* green bar that alerts us to the fact that our event was successfully created!



transactions_fraud_event

**Details**

Event type
transactions_fraud_event

Entity type
transaction

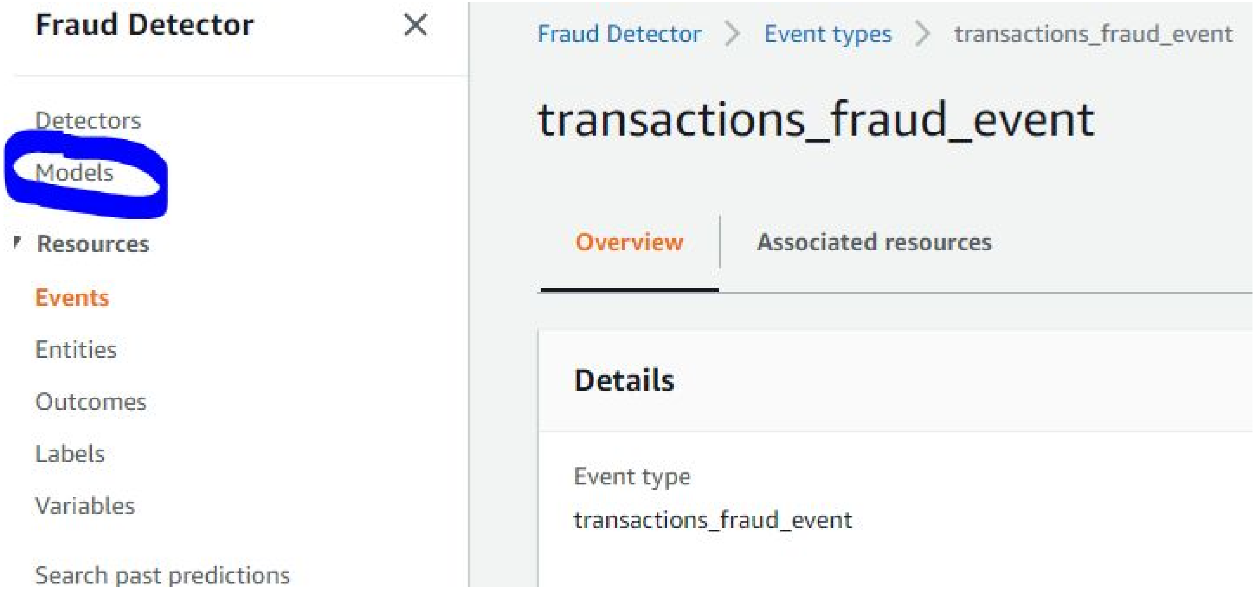
Date created
4 days ago

Last updated
4 days ago

**Variables** (9)

| Variable | Variable type |
|---|---|
| card4 | Custom - categorical |
| p_emaildomain | Custom - categorical |
| id_31 | Custom - categorical |
| transaction_amt | Total Order Price |
| id_29 | Custom - categorical |
| device_type | Custom - categorical |
| card6 | Custom - categorical |
| device_info | Custom - categorical |

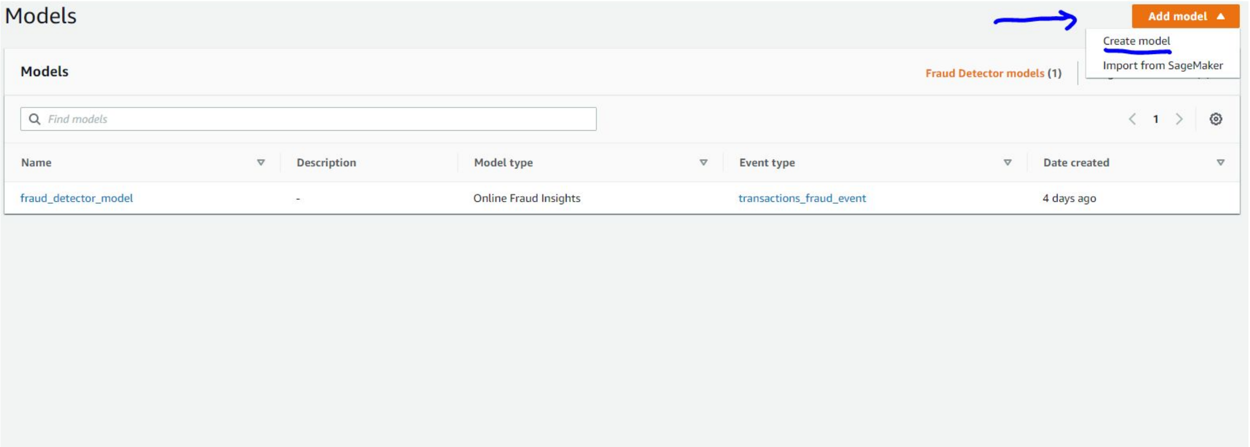
Now it’s time to create our Model.



transactions_fraud_event

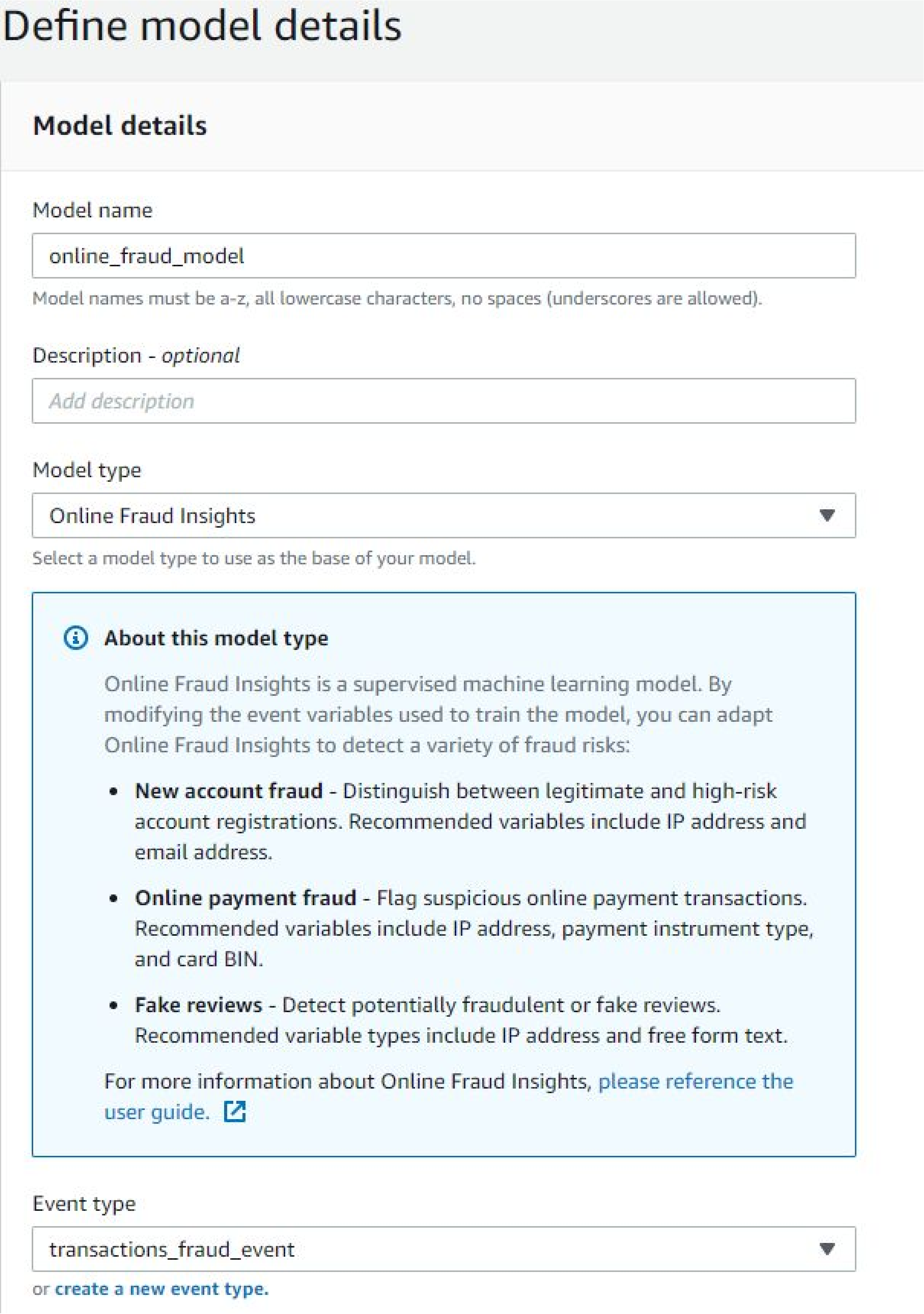
Details

Event type
transactions_fraud_event



| Name | Description | Model type | Event type | Date created |
|---|---|---|---|---|
| fraud_detector_model | - | Online Fraud Insights | transactions_fraud_event | 4 days ago |

Let’s take a moment to Define model details. We make sure to select our previously created *event type*.



# Define model details

## Model details

Model name

online_fraud_model

Model names must be a-z, all lowercase characters, no spaces (underscores are allowed).

Description - *optional*

*Add description*

Model type

Online Fraud Insights ▼

Select a model type to use as the base of your model.

**About this model type**

Online Fraud Insights is a supervised machine learning model. By modifying the event variables used to train the model, you can adapt Online Fraud Insights to detect a variety of fraud risks:

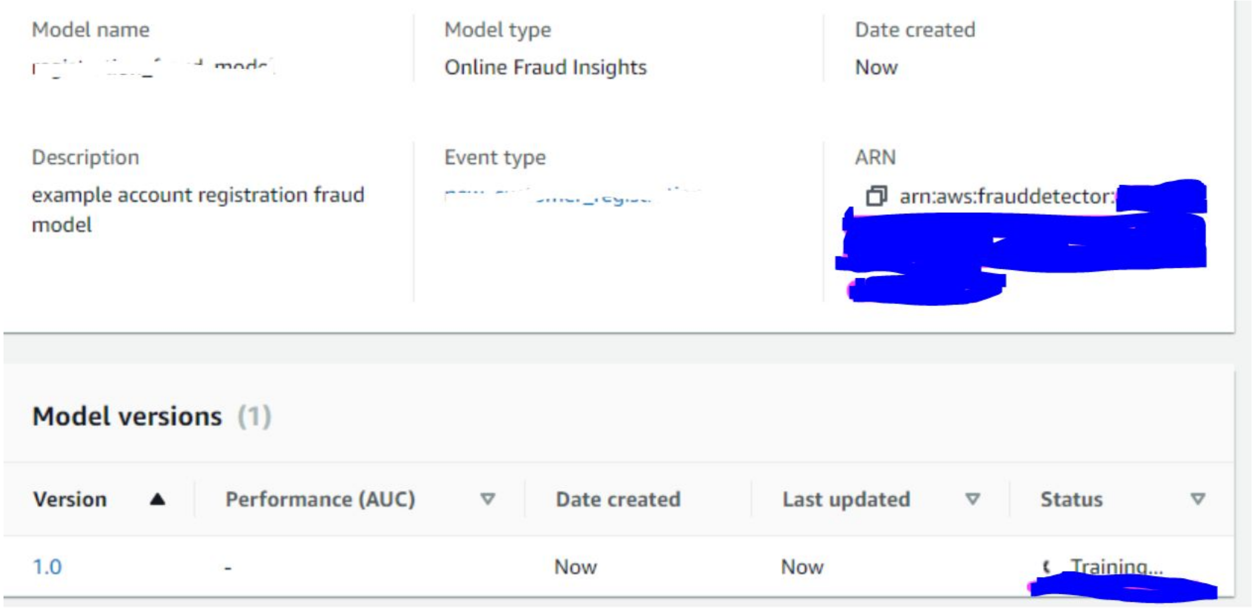
- **New account fraud** - Distinguish between legitimate and high-risk account registrations. Recommended variables include IP address and email address.
- **Online payment fraud** - Flag suspicious online payment transactions. Recommended variables include IP address, payment instrument type, and card BIN.
- **Fake reviews** - Detect potentially fraudulent or fake reviews. Recommended variable types include IP address and free form text.

For more information about Online Fraud Insights, please reference the user guide.

Event type

transactions_fraud_event ▼

or **create a new event type.**



Model name
[redacted]

Model type
Online Fraud Insights

Date created
Now

Description
example account registration fraud model

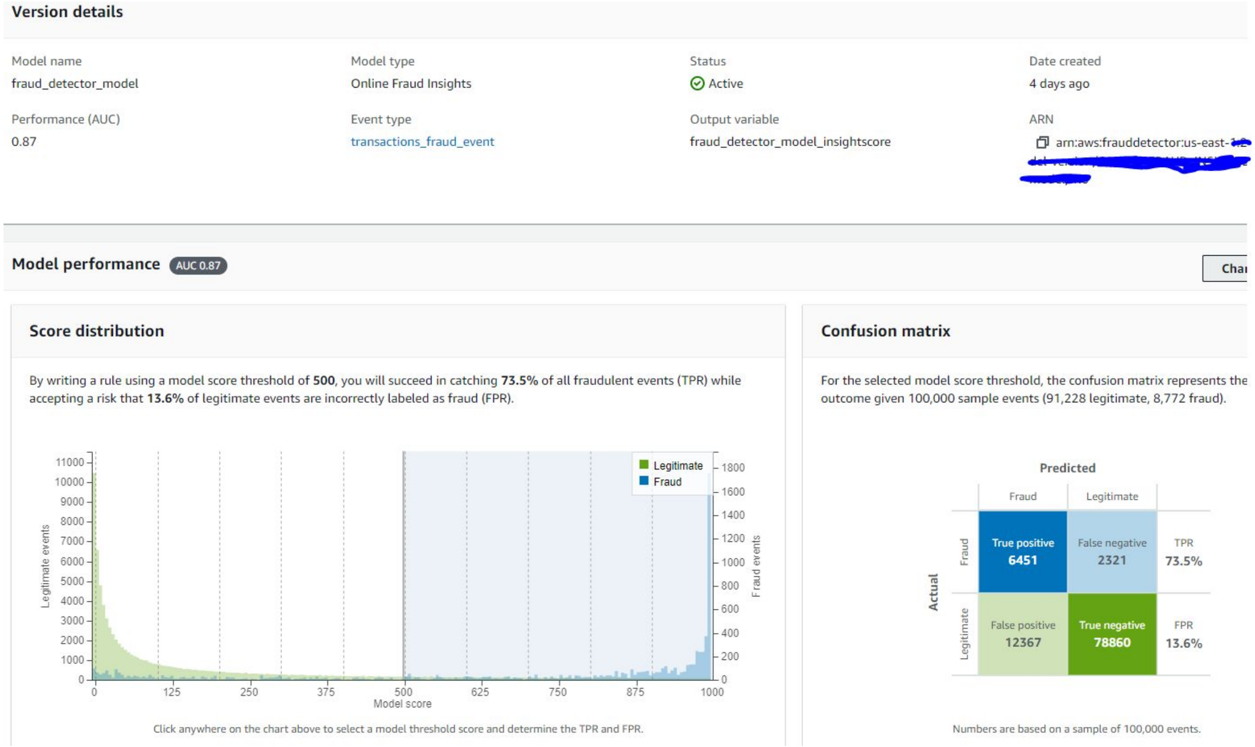
Event type
[redacted]

ARN
arn:aws:frauddetector:[redacted]

**Model versions** (1)

| Version ▲ | Performance (AUC) ▽ | Date created | Last updated ▽ | Status ▽ |
|---|---|---|---|---|
| 1.0 | - | Now | Now | Training... |

You can also check out your model's performance metrics!



**Version details**

Model name
fraud_detector_model

Model type
Online Fraud Insights

Status
✓ Active

Date created
4 days ago

Performance (AUC)
0.87

Event type
transactions_fraud_event

Output variable
fraud_detector_model_insightscore

ARN
arn:aws:frauddetector:us-east-[redacted]

**Model performance** AUC 0.87

**Score distribution**

By writing a rule using a model score threshold of **500**, you will succeed in catching **73.5%** of all fraudulent events (TPR) while accepting a risk that **13.6%** of legitimate events are incorrectly labeled as fraud (FPR).

Click anywhere on the chart above to select a model threshold score and determine the TPR and FPR.

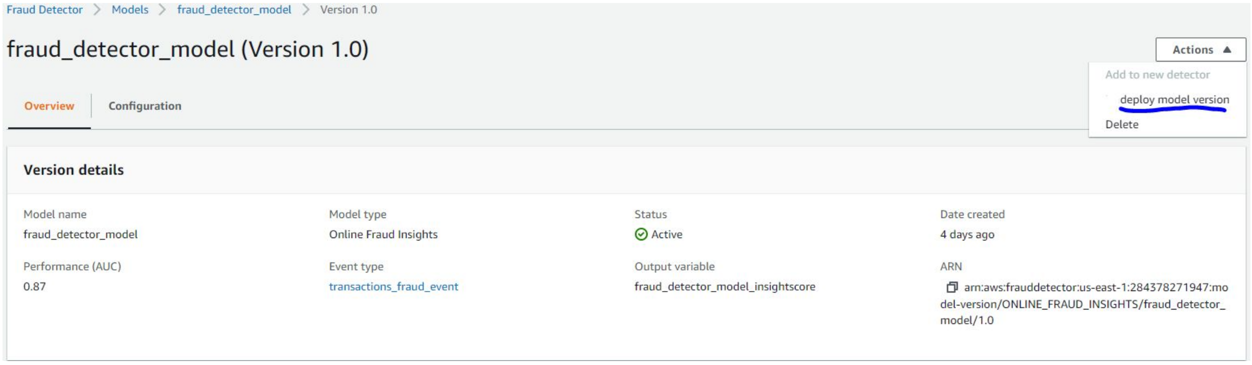
**Confusion matrix**

For the selected model score threshold, the confusion matrix represents the outcome given 100,000 sample events (91,228 legitimate, 8,772 fraud).

| | | Predicted | | |
|---|---|---|---|---|
| | | Fraud | Legitimate | |
| Actual | Fraud | True positive<br>6451 | False negative<br>2321 | TPR<br>73.5% |
| | Legitimate | False positive<br>12367 | True negative<br>78860 | FPR<br>13.6% |

Numbers are based on a sample of 100,000 events.

We can now proceed to deploy our Model.



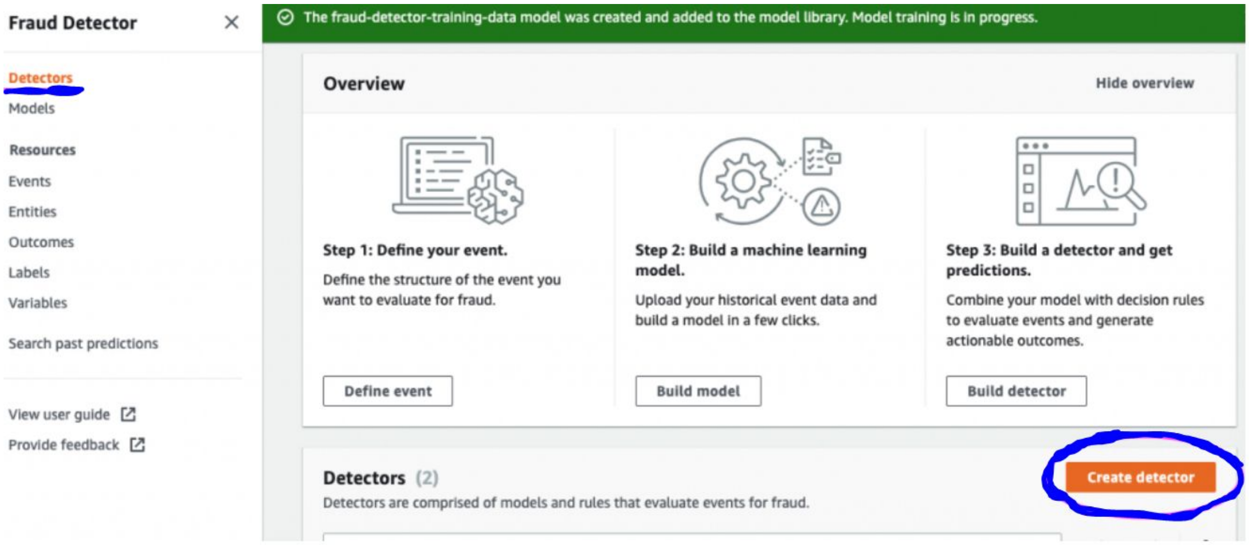
fraud_detector_model (Version 1.0)

Version details

| Model name | Model type | Status | Date created |
|---|---|---|---|
| fraud_detector_model | Online Fraud Insights | ✓ Active | 4 days ago |
| **Performance (AUC)** | **Event type** | **Output variable** | **ARN** |
| 0.87 | transactions_fraud_event | fraud_detector_model_insightscore | arn:aws:frauddetector:us-east-1:284378271947:model-version/ONLINE_FRAUD_INSIGHTS/fraud_detector_model/1.0 |

It’s time to generate real-time fraud predictions! At this point you have a deployed model that you’re happy with and want to use to get predictions.

We must build a Detector, which is a container for your models and rules. It’s your detection logic that you want to apply to evaluate the event.



The fraud-detector-training-data model was created and added to the model library. Model training is in progress.

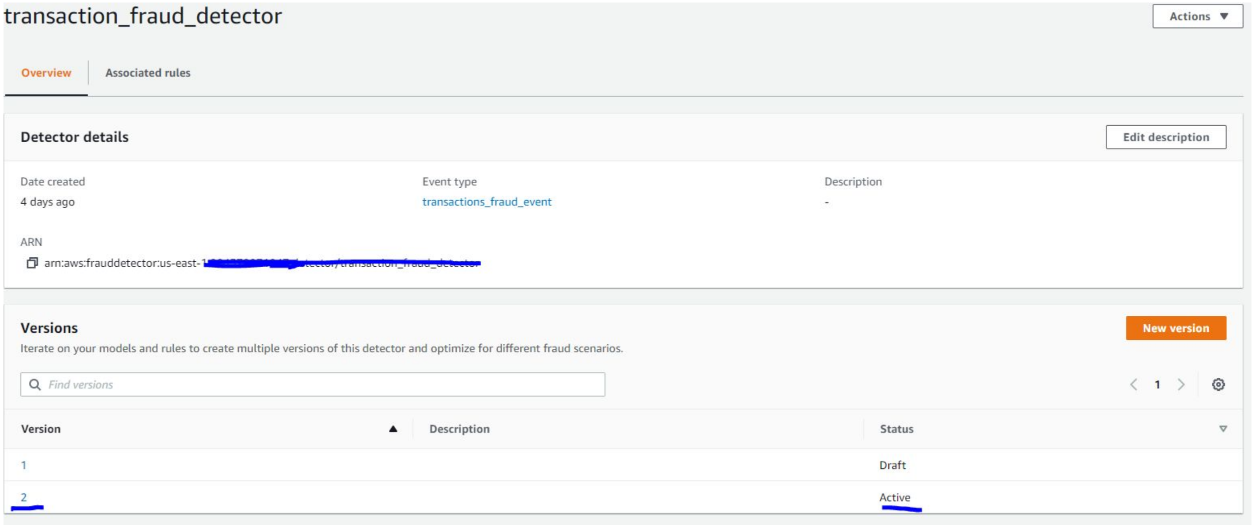
Overview

Step 1: Define your event.
Define the structure of the event you want to evaluate for fraud.

Step 2: Build a machine learning model.
Upload your historical event data and build a model in a few clicks.

Step 3: Build a detector and get predictions.
Combine your model with decision rules to evaluate events and generate actionable outcomes.

Detectors (2)
Detectors are comprised of models and rules that evaluate events for fraud.



transaction_fraud_detector

**Detector details**

Date created: 4 days ago

Event type: transactions_fraud_event

Description: -

ARN: arn:aws:frauddetector:us-east-[redacted]

**Versions**

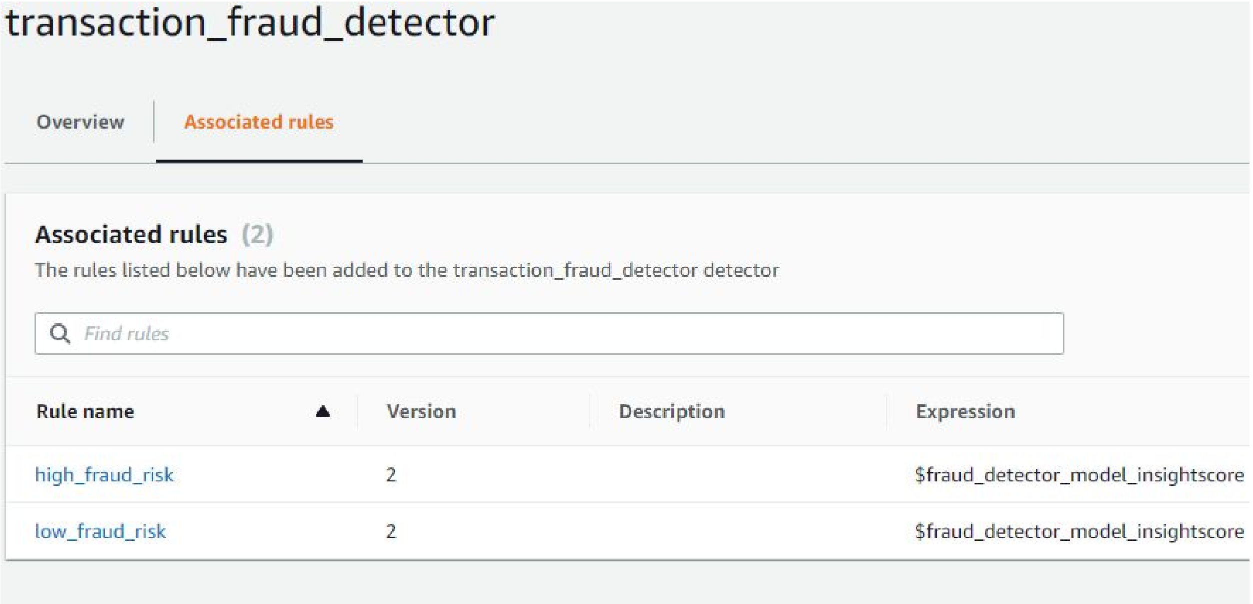
Iterate on your models and rules to create multiple versions of this detector and optimize for different fraud scenarios.

| Version | Description | Status |
|---|---|---|
| 1 | | Draft |
| 2 | | Active |

Step 4: Create rules to either accept, review, or collect more information based on model predictions.

We move on to establish some threshold rules.

The rules interpret the output of the *Model*. They also determine the output of the *Detector*.



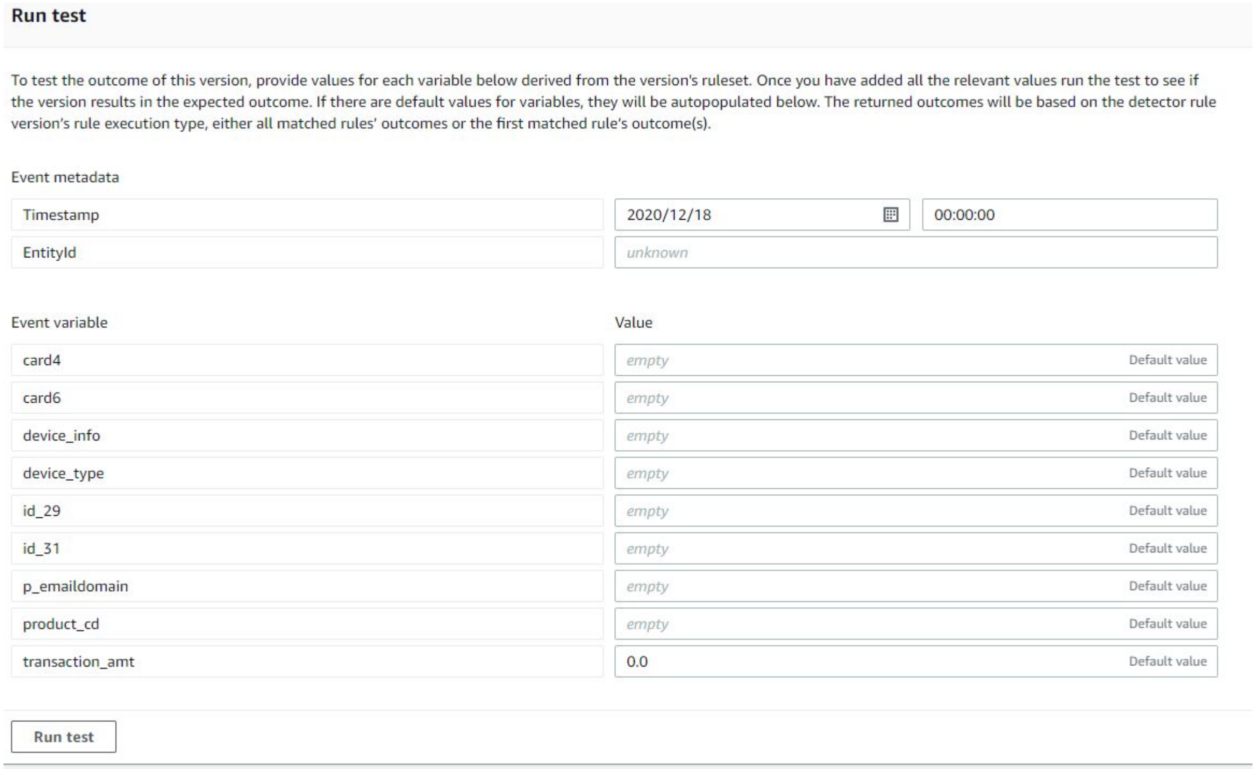
transaction_fraud_detector

**Associated rules (2)**

The rules listed below have been added to the transaction_fraud_detector detector

| Rule name | Version | Description | Expression |
|---|---|---|---|
| high_fraud_risk | 2 | | $fraud_detector_model_insightscore |
| low_fraud_risk | 2 | | $fraud_detector_model_insightscore |

If we go back to the Overview tab, we can even run a quick test! We can run tests to sample the *output* from our Detector.



**Run test**

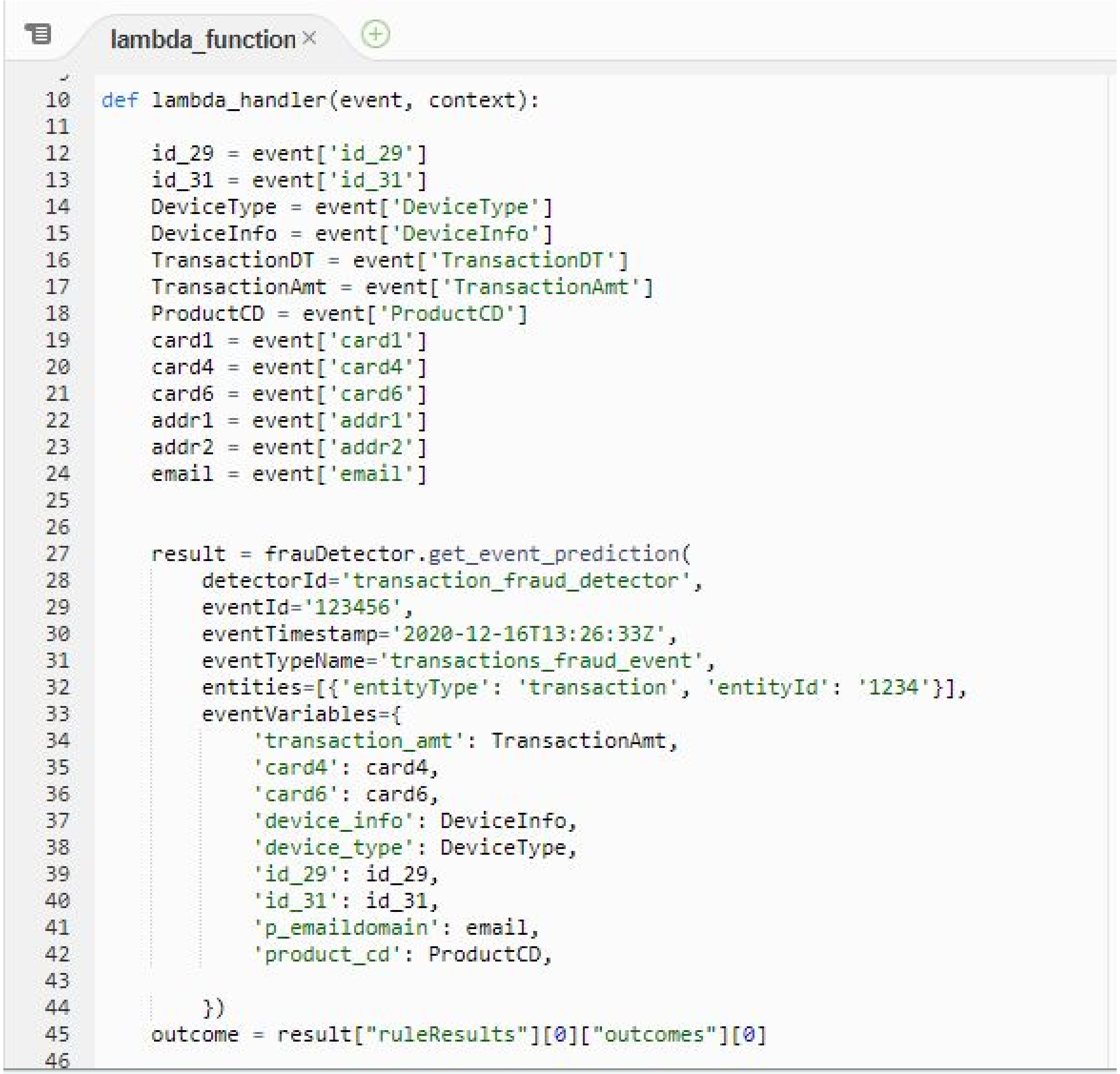
To test the outcome of this version, provide values for each variable below derived from the version's ruleset. Once you have added all the relevant values run the test to see if the version results in the expected outcome. If there are default values for variables, they will be autopopulated below. The returned outcomes will be based on the detector rule version's rule execution type, either all matched rules' outcomes or the first matched rule's outcome(s).

Event metadata

| | | |
|---|---|---|
| Timestamp | 2020/12/18 | 00:00:00 |
| EntityId | *unknown* | |

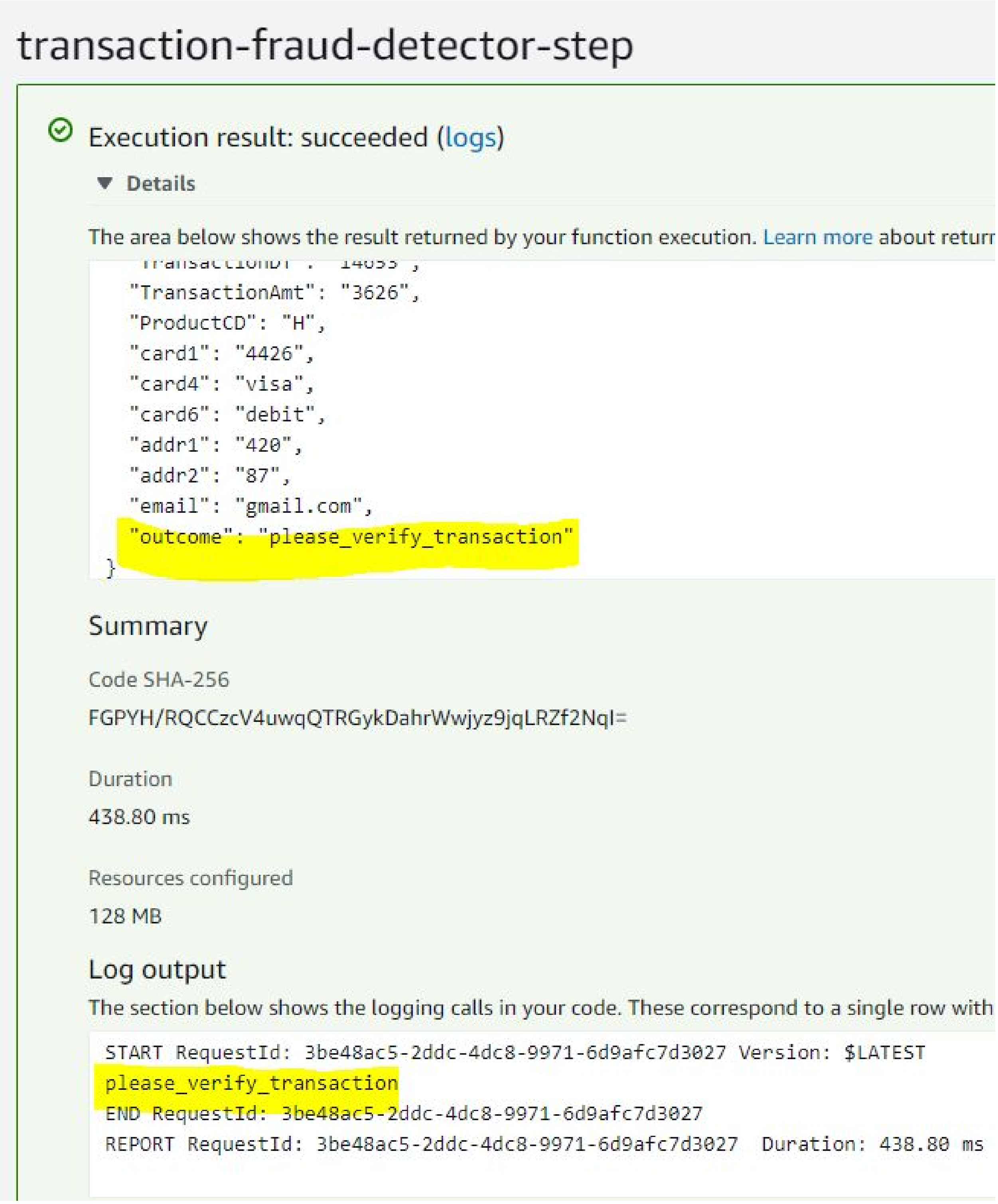
| Event variable | Value | |
|---|---|---|
| card4 | *empty* | Default value |
| card6 | *empty* | Default value |
| device_info | *empty* | Default value |
| device_type | *empty* | Default value |
| id_29 | *empty* | Default value |
| id_31 | *empty* | Default value |
| p_emaildomain | *empty* | Default value |
| product_cd | *empty* | Default value |
| transaction_amt | 0.0 | Default value |

Step 5: Call the Amazon Fraud Detector API from your online application to receive real-time fraud predictions and take action based on your configured detection rules. *(Example: an ecommerce application can send defined variables and receive a fraud score as well as the output from your rule)*



```
def lambda_handler(event, context):

    id_29 = event['id_29']
    id_31 = event['id_31']
    DeviceType = event['DeviceType']
    DeviceInfo = event['DeviceInfo']
    TransactionDT = event['TransactionDT']
    TransactionAmt = event['TransactionAmt']
    ProductCD = event['ProductCD']
    card1 = event['card1']
    card4 = event['card4']
    card6 = event['card6']
    addr1 = event['addr1']
    addr2 = event['addr2']
    email = event['email']


    result = frauDetector.get_event_prediction(
        detectorId='transaction_fraud_detector',
        eventId='123456',
        eventTimestamp='2020-12-16T13:26:33Z',
        eventTypeName='transactions_fraud_event',
        entities=[{'entityType': 'transaction', 'entityId': '1234'}],
        eventVariables={
            'transaction_amt': TransactionAmt,
            'card4': card4,
            'card6': card6,
            'device_info': DeviceInfo,
            'device_type': DeviceType,
            'id_29': id_29,
            'id_31': id_31,
            'p_emaildomain': email,
            'product_cd': ProductCD,

        })
    outcome = result["ruleResults"][0]["outcomes"][0]

```

We now have confirmation that you can call this Detector in real time and get your Fraud Predictions.



# transaction-fraud-detector-step

Execution result: succeeded (logs)

▼ Details

The area below shows the result returned by your function execution. Learn more about return

```
  "TransactionDT": "14033",
  "TransactionAmt": "3626",
  "ProductCD": "H",
  "card1": "4426",
  "card4": "visa",
  "card6": "debit",
  "addr1": "420",
  "addr2": "87",
  "email": "gmail.com",
  "outcome": "please_verify_transaction"
}
```

## Summary

Code SHA-256

FGPYH/RQCCzcV4uwqQTRGykDahrWwjyz9jqLRZf2NqI=

Duration

438.80 ms

Resources configured

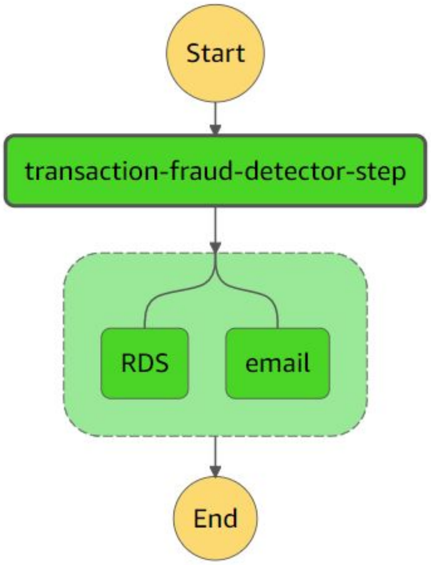
128 MB

## Log output

The section below shows the logging calls in your code. These correspond to a single row with

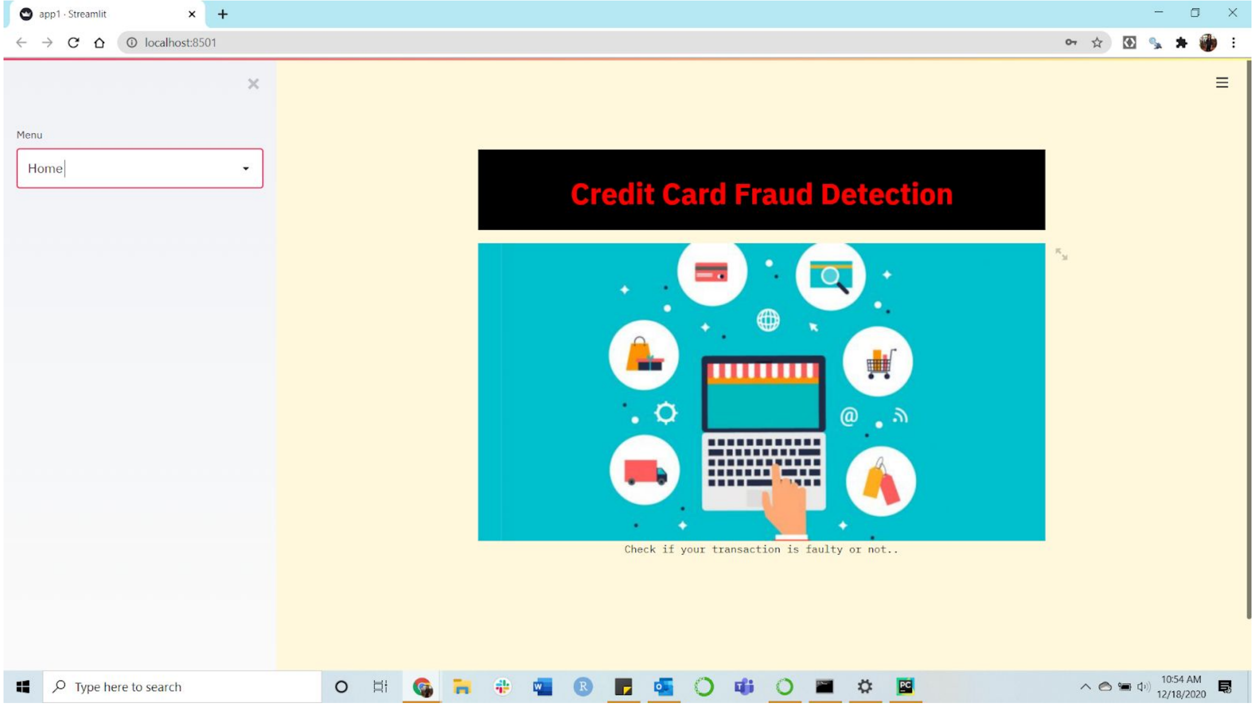
```
START RequestId: 3be48ac5-2ddc-4dc8-9971-6d9afc7d3027 Version: $LATEST
please_verify_transaction
END RequestId: 3be48ac5-2ddc-4dc8-9971-6d9afc7d3027
REPORT RequestId: 3be48ac5-2ddc-4dc8-9971-6d9afc7d3027  Duration: 438.80 ms
```

**Step Function:**



# Frontend: Streamlit Web App

**Page 1: Home page**



Menu

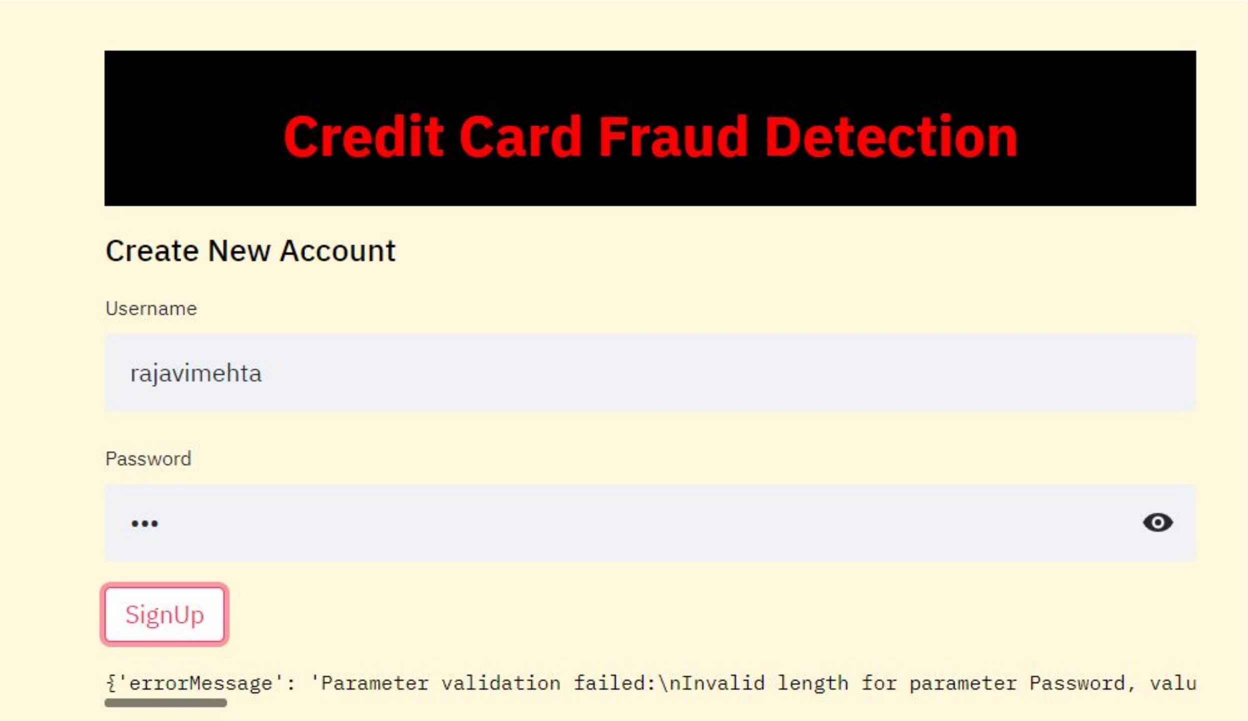
Home

Credit Card Fraud Detection

Check if your transaction is faulty or not..

**Page 2: Sign Up**

Below we have shown that if the password length is less than 6, it will throw an error and will not let the user create a new account



Credit Card Fraud Detection

Create New Account

Username

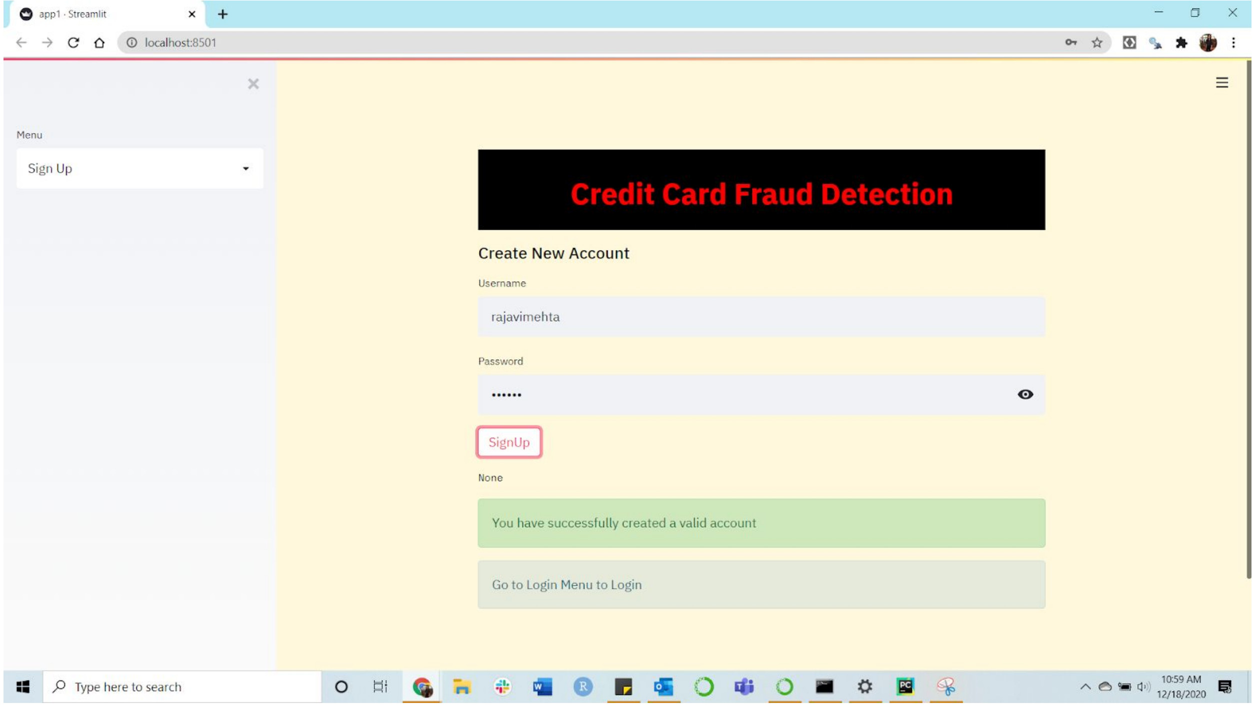
rajavimehta

Password

•••

```
{'errorMessage': 'Parameter validation failed:\nInvalid length for parameter Password, valu
```

Once, the password length is correct, it will create a new account as shown below



Credit Card Fraud Detection

Create New Account

Username

rajavimehta

Password

••••••

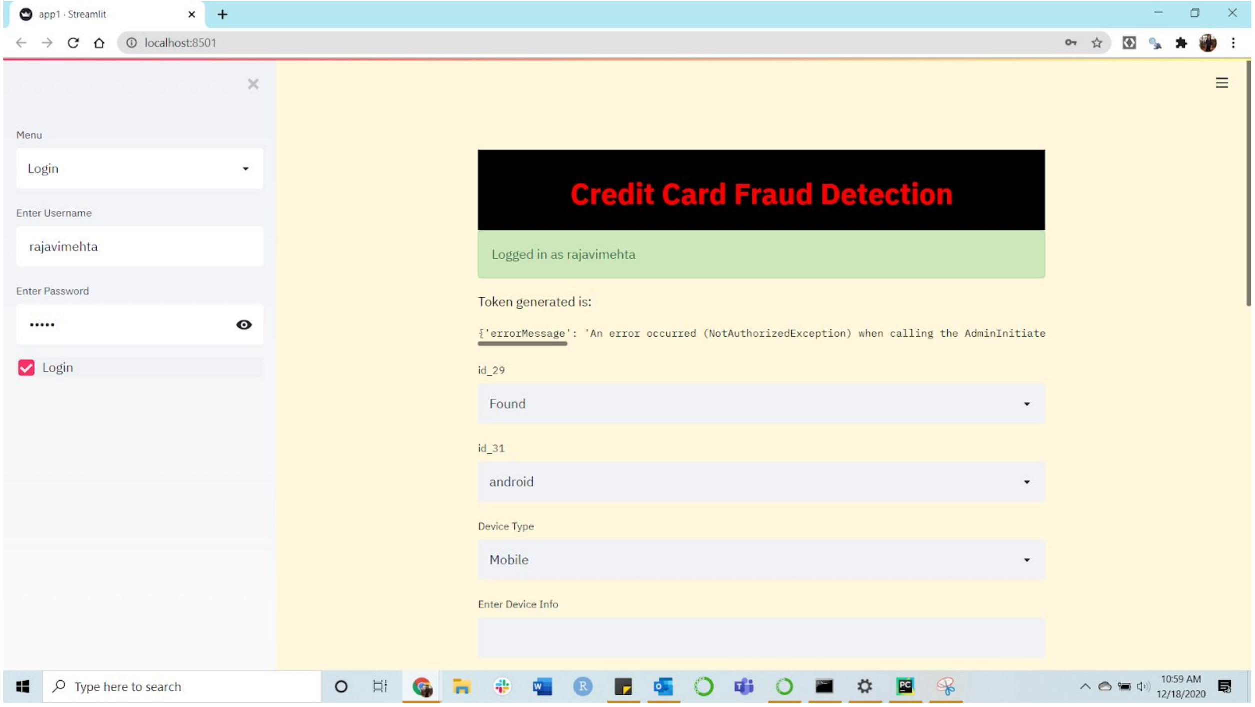
None

You have successfully created a valid account

Go to Login Menu to Login

**Page 3:**

When I tried logging with the wrong id(the one where my password was not satisfying the criterias and the account did not get created), the app threw the error that the user is not authorized to login



Menu

Login

Enter Username

rajavimehta

Enter Password

•••••

☑ Login

Credit Card Fraud Detection

Logged in as rajavimehta

Token generated is:

```
{'errorMessage': 'An error occurred (NotAuthorizedException) when calling the AdminInitiate
```

id_29

Found

id_31

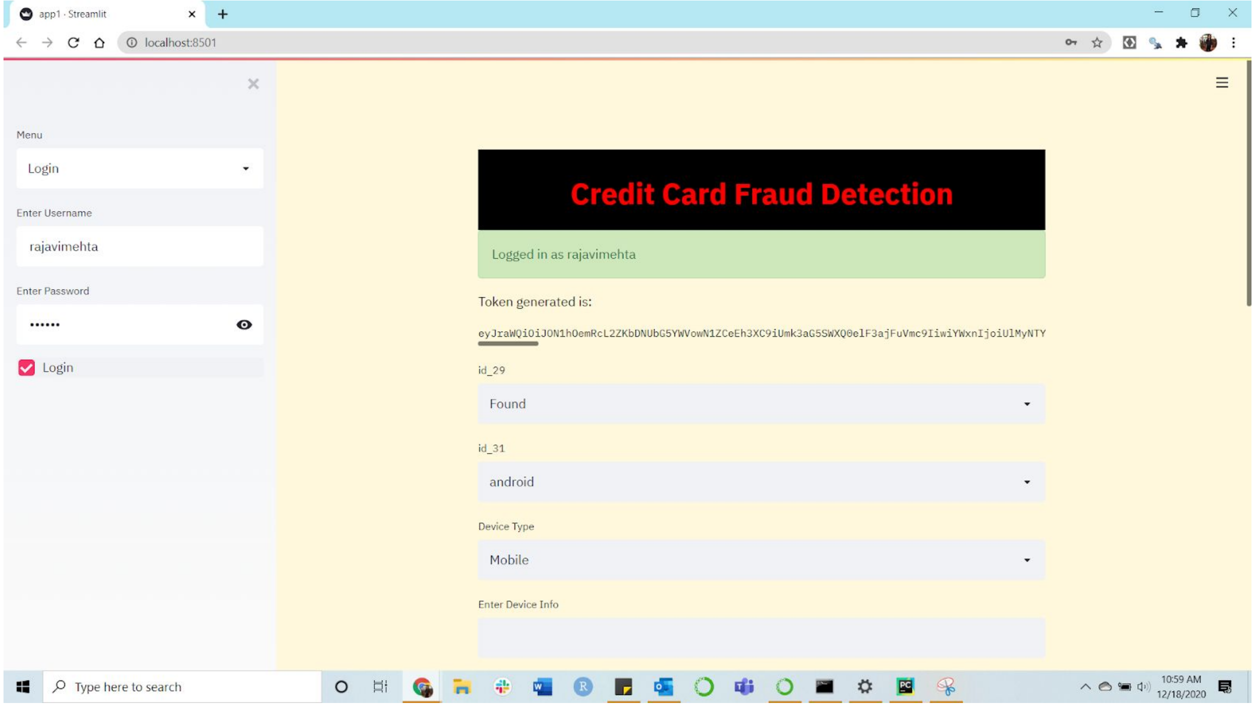
android

Device Type

Mobile

Enter Device Info

Once the user adds the correct credentials, a token is generated and the user can login and fill in the form as shown below:



Menu

Login

Enter Username

rajavimehta

Enter Password

••••••

☑ Login

Credit Card Fraud Detection

Logged in as rajavimehta

Token generated is:

```
eyJraWQiOiJON1hOemRcL2ZKbDNUbG5YWVowN1ZCeEh3XC9iUmk3aG5SWXQ0elF3ajFuVmc9IiwiYWxnIjoiUlMyNTY
```

id_29

Found

id_31

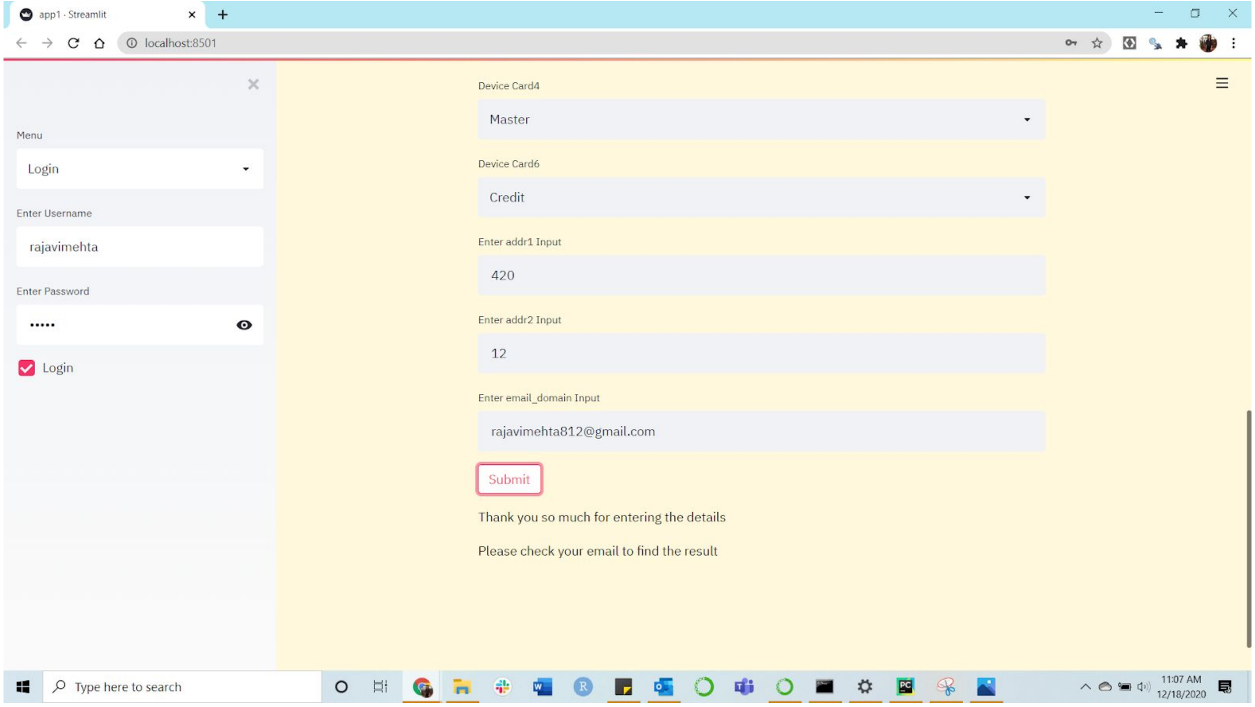
android

Device Type

Mobile

Enter Device Info

Once the user enters the details in the form, the user is informed that their details have been taken and they can check their email id to see if the transaction is faulty or not



Menu

Login

Enter Username

rajavimehta

Enter Password

•••••

☑ Login

Device Card4

Master

Device Card6

Credit

Enter addr1 Input

420

Enter addr2 Input

12

Enter email_domain Input

rajavimehta812@gmail.com

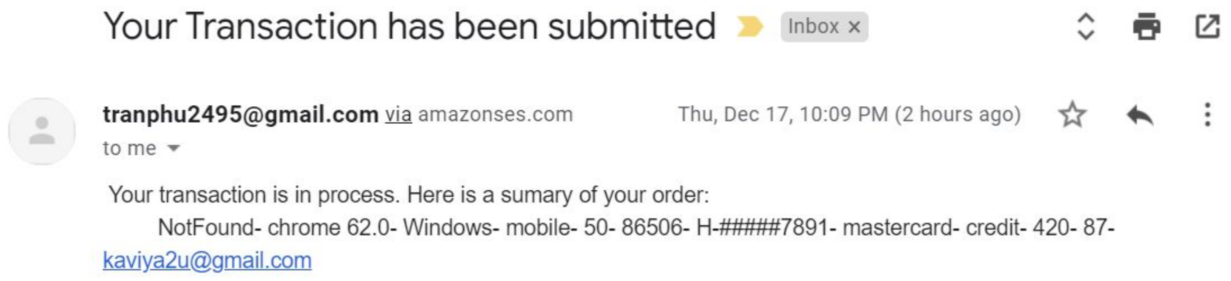
Submit

Thank you so much for entering the details

Please check your email to find the result

After the user submits their details successfully, they receive an email approving the receival of their credentials and a second email stating the output about the transaction being faulty or not

**Transaction Submitted Email:**

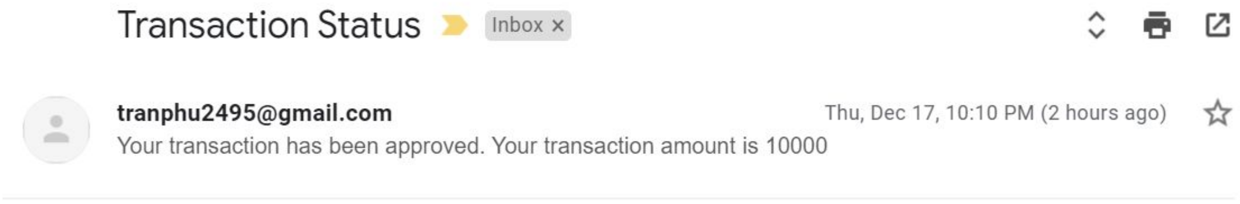


tranphu2495@gmail.com via amazonses.com
to me
Thu, Dec 17, 10:09 PM (2 hours ago)

Your transaction is in process. Here is a sumary of your order:

NotFound- chrome 62.0- Windows- mobile- 50- 86506- H-#####7891- mastercard- credit- 420- 87- kaviya2u@gmail.com

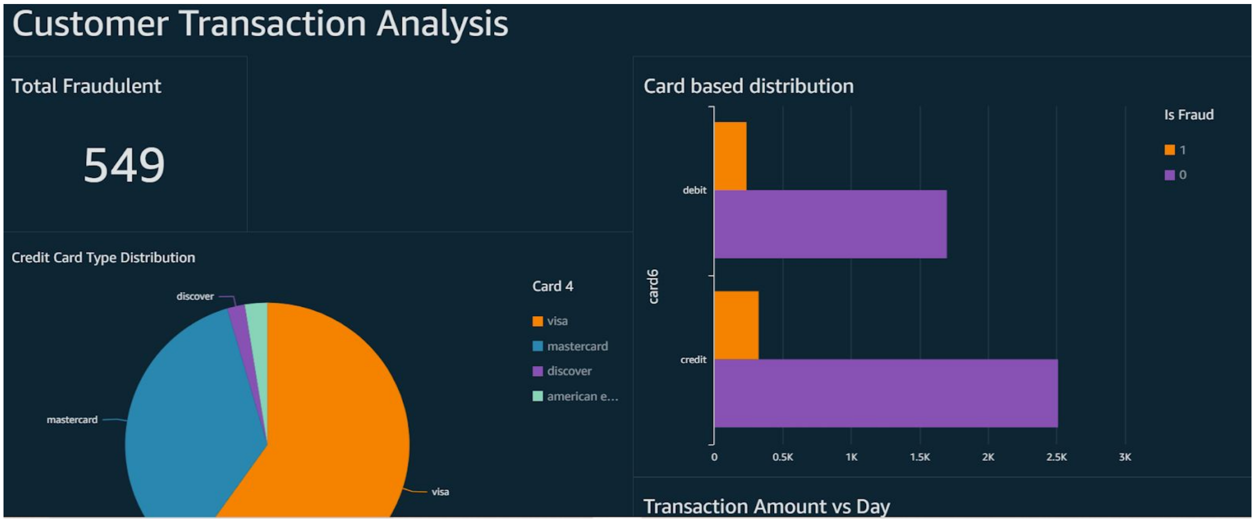
**Transaction Approval Email:**

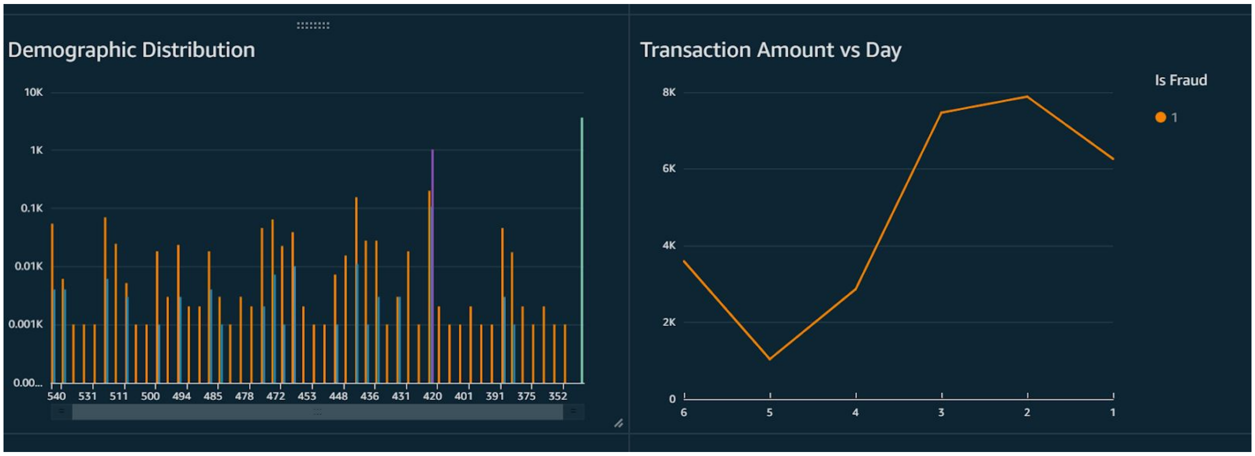


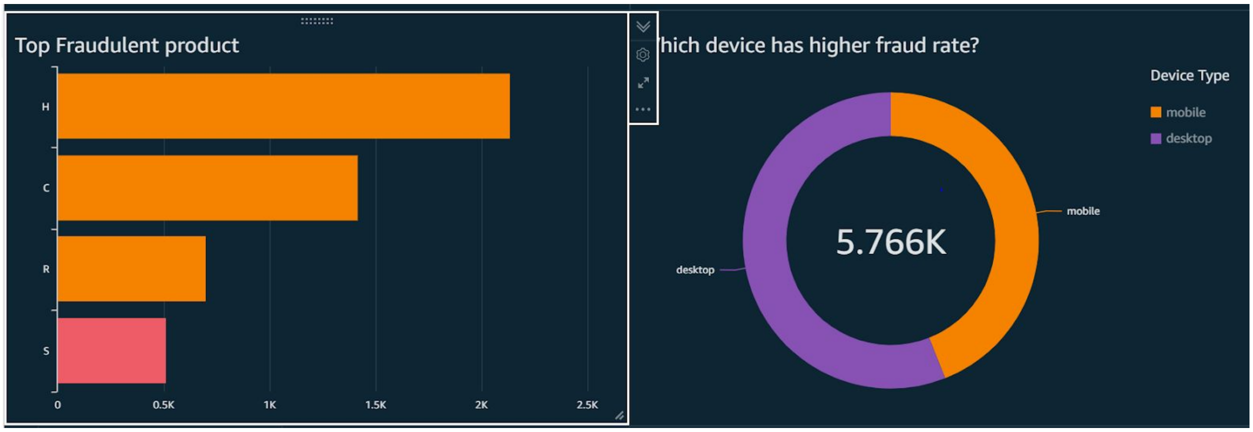
tranphu2495@gmail.com
Thu, Dec 17, 10:10 PM (2 hours ago)

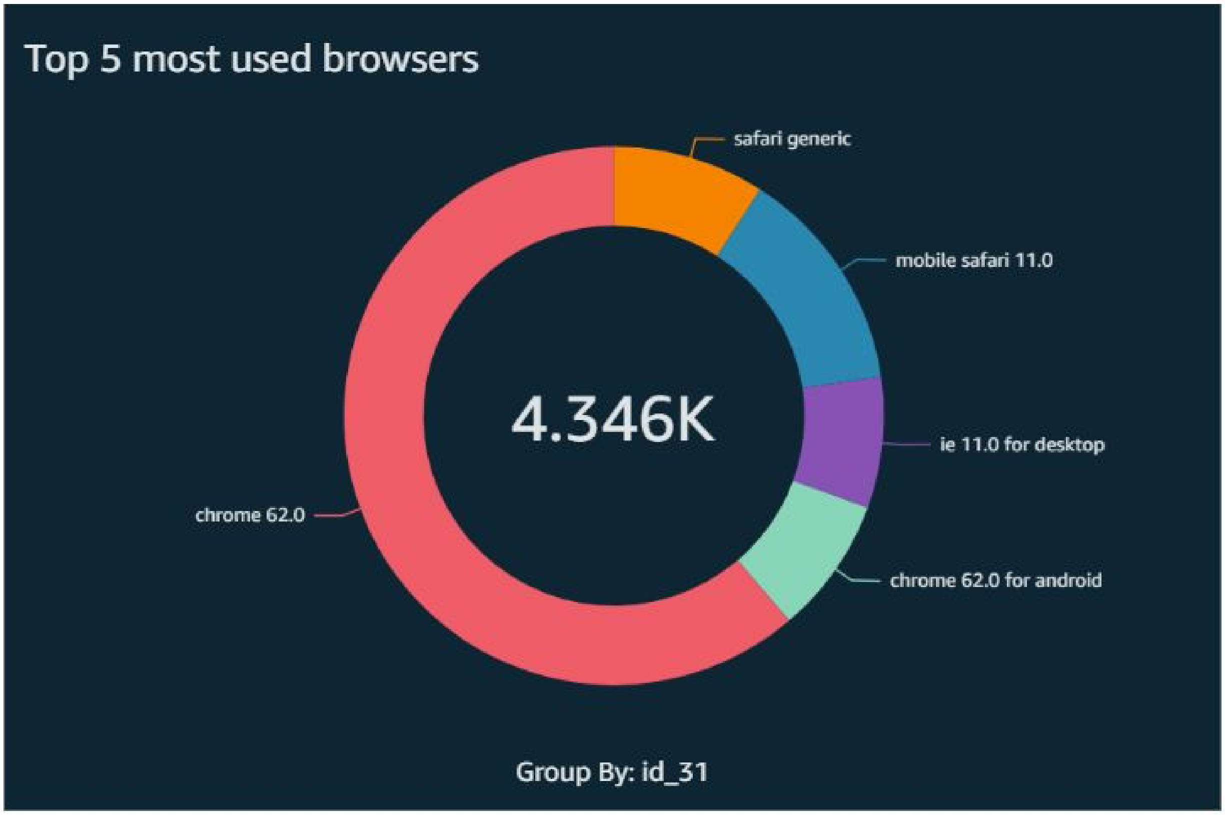
Your transaction has been approved. Your transaction amount is 10000

# Quicksight: Graphs

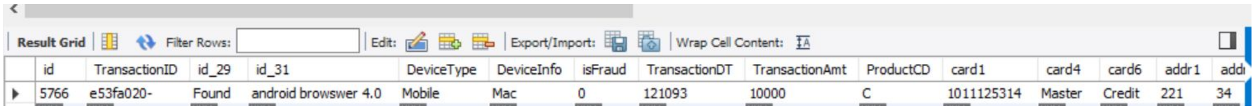



Demographic Distribution
10K
1K
0.1K
0.01K
0.001K
0.00...
540
531
511
500
494
485
478
472
453
448
436
431
420
401
391
375
352
Transaction Amount vs Day
Is Fraud
● 1
8K
6K
4K
2K
0
6
5
4
3
2
1


Top Fraudulent product
H
C
R
S
0
0.5K
1K
1.5K
2K
2.5K
hich device has higher fraud rate?
Device Type
mobile
desktop
5.766K
mobile
desktop



**Record being inserted into RDS:**



| id | TransactionID | id_29 | id_31 | DeviceType | DeviceInfo | isFraud | TransactionDT | TransactionAmt | ProductCD | card1 | card4 | card6 | addr1 | add |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5766 | e53fa020- | Found | android browswer 4.0 | Mobile | Mac | 0 | 121093 | 10000 | C | 1011125314 | Master | Credit | 221 | 34 |