Risk EYE

Revolutionizing Fraud Detection in Online Payments

01

Identified gaps in existing fraud detection systems for online payments.

Our goal: Build a real-time, efficient fraud detection system to safeguard users and businesses from fraudulent transactions.

Inspired by real-world cases and the need for scalable fraud prevention solutions.

Inspiration

02

What It Does?

RiskyEye provides real-time detection of fraudulent payment activities.

Analyzes key transaction details (amount, sender, recipient, location) to flag anomalies.
Generates instant fraud alerts and detailed reports for businesses.

Supports integration with payment gateways for seamless protection.

03

Tech Stacks used :

Postman API
Flask
Ml Algorithms: Numpy, Panda, Scborn

How We Built It

Where this feature will be implemented?

1. **Mobile Banking Apps / Payment Apps**
   (e.g., Paytm, Google Pay, or Revolut)
2. **E-commerce Platforms**
   (as part of payment gateways like Razorpay or Stripe)
3. **Bank APIs and Payment Gateways**
   (e.g., during card payments or account transfers)

How a User Interacts with the Fraud Detection Feature

Transaction Initiation

The user logs into their banking or payment app and initiates a payment.
-Example: A user tries to transfer ₹10,000 to another account using the mobile app.

Suspicious Transaction Detected

The system sends the transaction data (amount, time, location, etc.) to the backend.
The fraud detection model checks whether the transaction looks legitimate or suspicious.

Transaction Validation in the Background

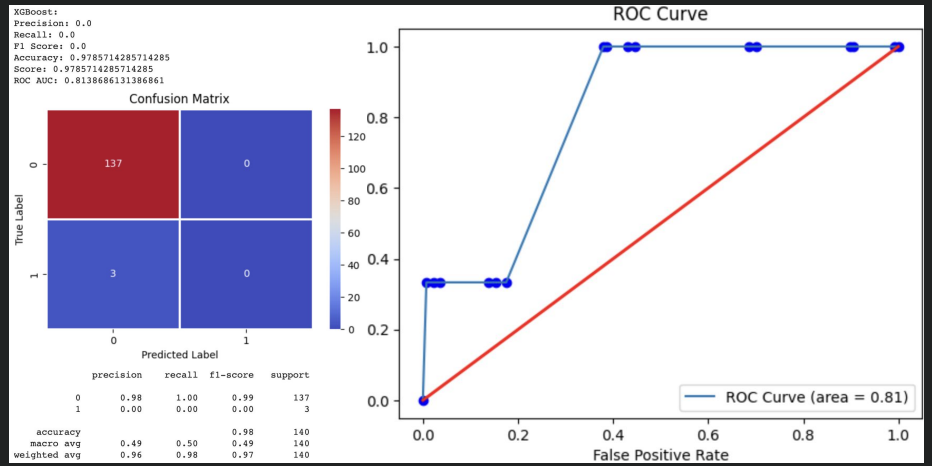
The system sends the transaction data (amount, time, location, etc.) to the backend.
The fraud detection model checks whether the transaction looks legitimate or suspicious.

Alert System

Alert system sends notifications through mails or messages if the fraud score exceeds a threshold. Its available for manual review afterwards

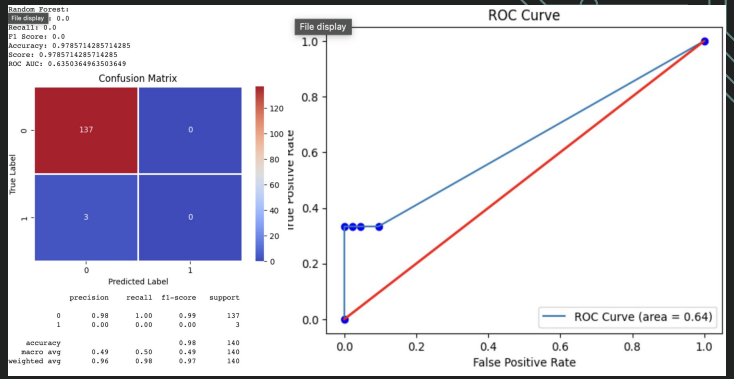
ML MODEL:

XGBoost (Extreme Gradient Boosting : Our Model

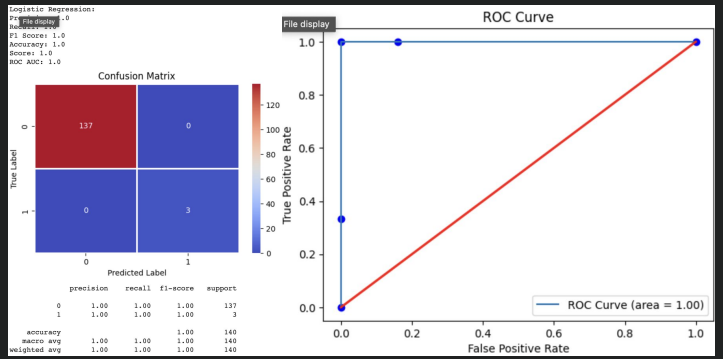


We went with XG Boost Model for its accuracy over Random forest and Logistics Regression

Random Forest



Logistics Regression



Process Pipeline

Data Overview

Data Preprocessing

Exploratory data analysis

Feature Engineering

Model Training

Evaluation

Parameters

Transaction_id : Unique identifier for tracking individual transactions.
Transfer_type : Specifies the type of transaction, like "transfer" or "cash out."
Amount : The total value being transferred in the transaction.
Sender_account: The account initiating the transaction.
Initial_balance_p1: The balance of the sender's account before the transaction.
New_balance_1 : The balance of the sender's account after the transaction.
Recipient_account : The account receiving the transaction amount.
Initial_balance_p2 : The balance of the recipient's account before the transaction.
New_balance_2 : The balance of the recipient's account after the transaction.
isFraud : Indicates whether the transaction is identified as fraudulent.

Parameters

isFlaggedFraud : Shows if the transaction was flagged as potentially fraudulent.
Transaction_Date : The date on which the transaction occurred.
Transaction_Time : The time when the transaction took place.
Coordinates : Location data associated with the transaction.
Ip_addresses : The IP address(es) involved in the transaction, helping to identify its origin.

What's Next for RiskyEye

Expanding to support more payment platforms and gateways.

Implementing user-facing fraud resolution tools. Refining our AI to detect increasingly sophisticated fraud techniques.

Aiming to bring RiskyEye to market for small and medium businesses.