Fraud Detection in Digital Advertising

Using SQL, Machine Learning, and Graph Theory

Deborah Gozlan - Fraud Data Analyst

August 2025

Data Science Portfolio Project

Executive Summary

This project detects fraudulent clicks and suspicious advertiser activity using SQL-first ingestion & cleaning, supervised and unsupervised machine learning, and graph theory for detecting fraudster communities.

The goal is to simulate real-world messy data, clean and prepare it using both SQL and Python, engineer fraud-relevant features, and apply multiple fraud detection techniques before presenting insights in Tableau/Looker dashboards.

Deliverables include:

- PostgreSQL database (raw, messy, cleaned data)
- SQL cleaning and KPI scripts
- Python notebooks with advanced cleaning & ML
- Tableau/Looker dashboards
- Excel data dictionary
- GitHub repository with professional documentation

Problem Statement

Digital advertising fraud causes billions in losses each year. Fraudsters use:

- Shared IP addresses to mask identities
- Bots to generate fake clicks
- Coordinated networks (fraud rings) to exploit platforms

The challenge:

- Detect fraudulent activity accurately
- Reduce false positives to avoid harming real advertisers
- Provide actionable insights for fraud prevention

Project Goal

To create an end-to-end fraud detection pipeline that:

- Ingests data into PostgreSQL before any processing
- Simulates realistic messy data
- Cleans and prepares it using SQL and Python
- Applies supervised & unsupervised ML for fraud detection
- Uses graph theory to detect fraud rings
- Builds interactive BI dashboards for fraud monitoring

Methodology

Data Ingestion:

- Load Kaggle TalkingData dataset directly into PostgreSQL
- Create relational schema (raw_clicks, ads, ad_performance, ad_connections)

Messy Data Simulation:

Why?

In real-world scenarios, data rarely arrives clean. It often comes from:

- Multiple systems with different formats
- Manual customer entry, causing typos & inconsistencies
- Different departments using their own conventions
- Input requiring clarification from subject matter experts (SMEs)

How?

We will introduce:

- Extra spaces in text fields
- Mixed casing in categorical values
- Mixed date formats
- Numbers stored as text
- Comma vs dot decimal separators
- Special characters in text
- Missing values in key fields
- Duplicates with variations
- Outlier values
- Additional messy variables (contact_email, customer_phone, notes)

Advanced Regex Example (Email Cleaning)

```
import re


def clean_email(email):
    if not isinstance(email, str):
        return None
    email = email.strip().lower()
    email = re.sub(r"\s*\(at\)\s*|\s*\[at\]\s*", "@", email)
    email = re.sub(r"\s*@\s*", "@", email)
    email = re.sub(r"[^a-z0-9@._-]+", "", email)
    if not re.match(r"^[a-z0-9._%-]+@[a-z0-9.-]+\.[a-z]{2,}$", email):
        return None
    return email
```

Deliverables

- PostgreSQL database (raw, messy, cleaned data)

- SQL cleaning & KPI scripts

- Python ML notebooks

- Tableau/Looker dashboards

- Excel data dictionary

- GitHub repository (README + PDF)

14-Day Roadmap

Day 1-2: SQL ingestion & messy data simulation

Day 3-4: SQL cleaning & KPI queries

Day 5-6: Python cleaning

Day 7-8: Advanced cleaning (fuzzy, KNN)

Day 9-10: Supervised ML

Day 11-12: Unsupervised ML

Day 13: Graph theory

Day 14: Dashboards & documentation

Next Steps

- Day 1: Create PostgreSQL schema & import Kaggle data
- Day 2: Simulate messy data
- Day 3-4: Clean data in SQL & create fraud KPIs
- Day 5+: Python cleaning & feature engineering
- Day 8+: ML training & evaluation
- Day 13+: Dashboards & delivery