AAI-540 ML Design Document

CMS Open Payments Risk Scoring & Anomaly Detection

Team Info

Project Team Group #: 3
Authors: Swapnil Patil, Jamshed Nabizada and Tej Bahadur Singh
Business Name: CMS (Center of Medicare & Medicaid Services)
Publication Date: 01/19/2026

Team Workflows

GitHub Project Link: https://github.com/swapnilprakashpatil/aai540_3proj
Asana Board Link:
https://app.asana.com/1/952672460738672/project/1212851836514318/list/1212851844967962
Team Tracker Link: https://docs.google.com/document/d/1kD-dVUTuNQrCbTmJ_lkGSUqfJPb4cMDS53SMUyYDtwQ/edit?usp=sharing

Project Scope

Project Background:

The CMS Open Payments program publishes information about financial relationships between drug/medical device companies ("reporting entities") and healthcare providers ("covered recipients") to promote transparency. These relationships can include payments for items such as meals, travel, gifts, speaking fees, and research-related transfers of value. The published data is open to interpretation and does not inherently indicate an improper relationship (Centers for Medicare & Medicaid Services [CMS], 2025a).

This project builds an ML system that assigns a risk score to Open Payments records (or aggregated entities) to prioritize statistically unusual payment patterns for compliance review. The system is designed for triage/prioritization and will not label records as fraud. The ML problem is framed as unsupervised anomaly detection on large tabular data.

Technical Background:

How the model will be evaluated

Because Open Payments does not provide a “fraud” ground-truth label, evaluation focuses on ranking usefulness and stability:

- **Top K utility:** The top-ranked anomalies should represent truly unusual patterns. The system will measure concentration of unusual behavior in the top K results (e.g., amount spikes vs peers, unusually high payer diversity).
- **Temporal stability & drift:** Compare score distributions and anomaly rates over time to ensure stability and detect drift.
- **Qualitative sanity review:** Provide “reason codes” (e.g., peer deviation, spikes, unusual mix) for the highest-risk entities.

Data source and volume

Primary dataset: Open Payments Program Year 2024 (General Payments). PY2024 was published June 30, 2025, and covers payments made between January 1 and December 31, 2024. CMS reports PY2024 includes approximately **16.16 million records** totaling **$13.18B** in payments/transfers of value, and the site reflects the most recent seven years of data (CMS, 2025a).

Project-specific ingestion results: the extracted General Payments table contains **15,397,627 rows** and **91 columns** (computed from the downloaded dataset). The dataset’s size reinforces the need for parquet conversion, partitioning, and batch processing.

Data preparation

- Download program-year extract → store in S3 (raw)
- Convert CSV to **partitioned parquet** (program_year + month)
- Type normalization (dates, numeric amounts), missingness handling, and deduplication
- Standardize and validate categorical fields; enforce schema constraints based on the CMS data dictionary/methodology documentation (CMS, 2025b).

Exploratory analysis (EDA)

- Payment amount distributions (heavy-tailed, log-scale)
- Variation by provider attributes (specialty, state, covered recipient type)
- Diversity of payment categories (nature/form)
- Temporal and seasonality patterns aligned with annual reporting/publication cycles

Hypothesized main features

The model is expected to learn “unusualness” primarily from:

- Amount magnitude aggregates (sum/mean/max/std)

- Frequency (count of payments)
- Diversity metrics (distinct reporting entities paying the same recipient)
- Mix/entropy of “nature of payment” and “form of payment” categories
- Peer deviations against specialty + state peer groups (recipient vs similar providers)

Model type

- Baseline: robust peer-group outlier scores (median/IQR deviation)
- Primary model: Isolation Forest on aggregated features (CPU-friendly, strong tabular anomaly baseline)
- Optional comparator: Local Outlier Factor on sampled/aggregated data

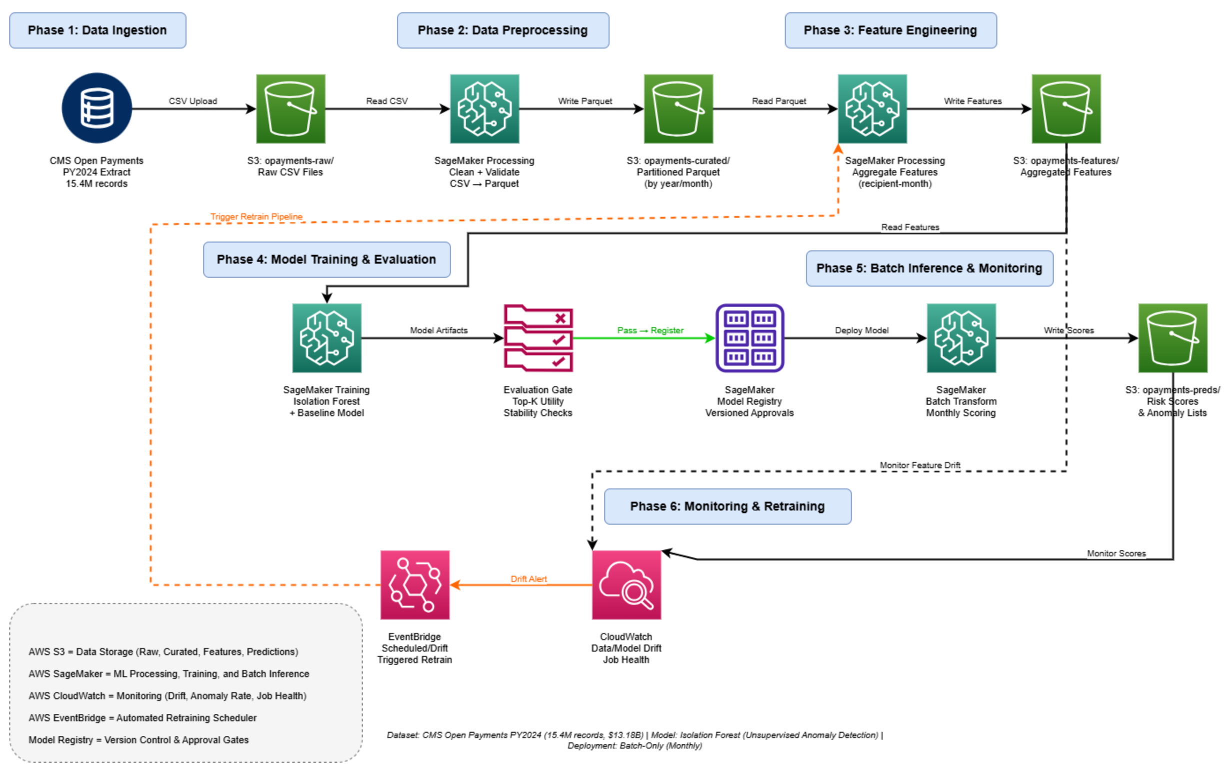
Goals vs non-goals:

Goals	Non-goals
Build an end-to-end AWS ML workflow: ingest → preprocess → feature engineering → train → registry → batch scoring.	Real-time streaming ingestion or real-time detection.
Generate interpretable risk scores and top K anomaly lists with “reason codes.”	Automated enforcement actions or definitive fraud claims.
Add monitoring for data drift and anomaly-rate drift; retrain on schedule or drift thresholds.	Production UI: outputs are batch files/tables and demo artifacts.
Keep AWS within limited credits using batch inference and small CPU jobs.	Linking external datasets to infer protected attributes or intent.
Maintain correct program framing: unusual pattern detection, not wrongdoing/fraud determination.	Maximizing state-of-the-art anomaly methods at the cost of simplicity and stability.

Solution Overview

The system ingests the PY2024 Open Payments general payments data into S3, converts it to curated parquet, engineer’s recipient-month aggregate features, trains an anomaly detection model, and produces batch risk scores. Monitoring checks for feature distribution drift and score/anomaly-rate drift, and retraining is triggered on schedule or drift thresholds. This batch-first design aligns well with the Open Payments annual publication lifecycle and refresh model (CMS, 2025a).

CMS Open Payments Anomaly Detection - ML Workflow



Data Sources:

CMS Open Payments Program Year 2024 public dataset (general payments focus) (CMS, 2025a). https://openpaymentsdata.cms.gov/datasets/download

- CMS summary: **~16.16M records** totaling **$13.18B** for PY2024 (CMS, 2025a).
- Project extract: **15,397,627 rows × 91 columns** (computed after ingestion).

Why this dataset?

- Real-world healthcare compliance/ethics transparency domain
- Large scale supports realistic pipeline engineering
- Annual refresh cycle supports drift + retraining storyline

Risks

- Potential **PII** exposure (provider identity/location); minimize use of direct identifiers in modeling.
- Interpretation risk: outputs must be framed as “unusual patterns,” consistent with CMS transparency guidance.

Data Engineering:

Storage

- s3://opayments-raw/ — raw downloads
- s3://opayments-curated/ — cleaned parquet (partitioned)
- s3://opayments-features/ — feature tables for training/scoring
- s3://opayments-preds/ — scored outputs

Preprocessing

- CSV → parquet conversion + partitioning
- Type casts + standardization
- Missing value handling
- Deduplication by record identifier fields per CMS data definitions.

Training Data:

Split strategy

Time-based split:

- Train on early months (or prior year)
- Validate on middle months
- Test on later months (or next year)

Labeling techniques

Weak evaluation signals from publication metadata such as changed vs unchanged records

Feature Engineering:

Fields to use / exclude

Use: amount, payment date, nature/form, reporting entity identifiers, specialty/taxonomy, state, recipient type.

Exclude: provider names and free text fields; avoid features that personalize to individuals.

Combinations / bucketing

- Aggregate to recipient-month
- Amount of log transforms
- Peer group normalization (specialty + state + recipient type)

Transformations

- log1p(amount)
- robust scaling (median/IQR)
- limited encoding for high-cardinality categories (frequency encoding)

Model Training & Evaluation:

Training method
Train Isolation Forest on aggregated feature table; tune contamination level to match review capacity (e.g., top 0.5–2%).

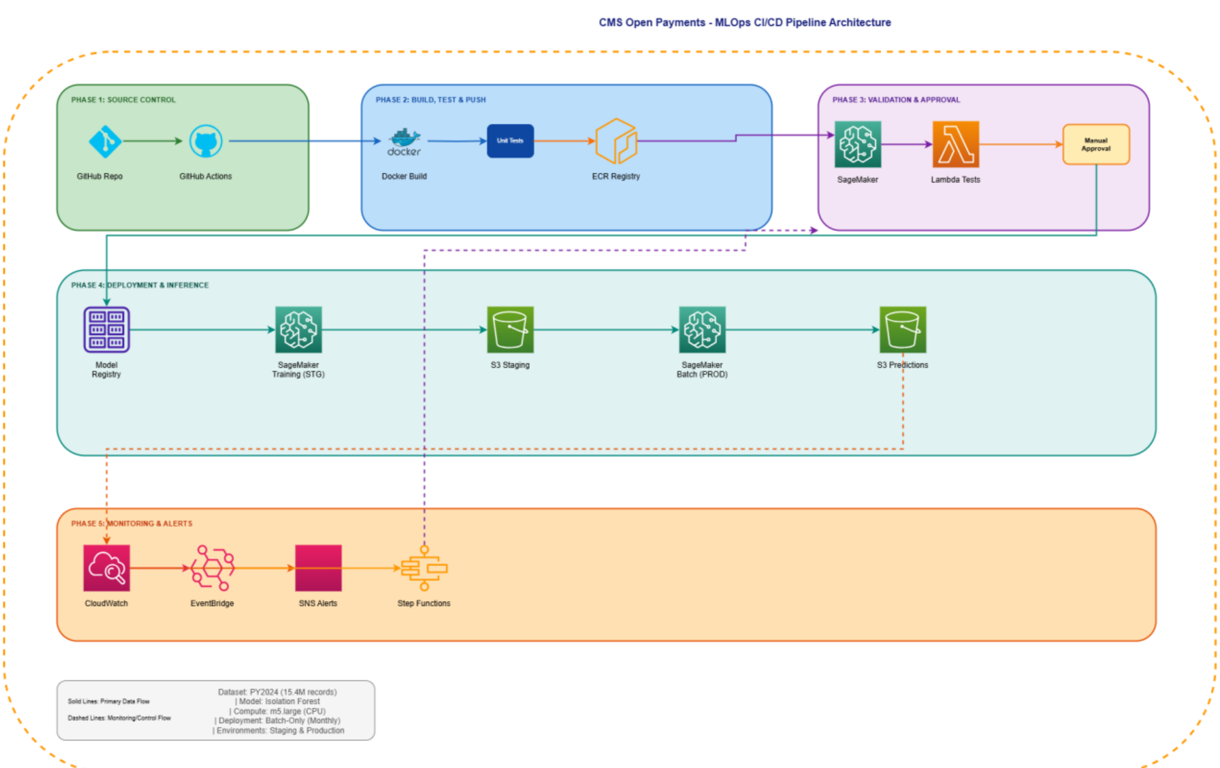
Algorithm
Isolation Forest + baseline robust peer outlier scoring.
Key parameters (initial)

- n_estimators: 200–500
- contamination: 0.005–0.02
- max_samples: 256 or auto
- max_features: 0.7–1.0

Evaluation

- Top K utility + stability checks
- Drift checks on features and scores
- Manual review of reason codes for top anomalies

Model Deployment:



Instance size
Small CPU instances for processing/training/batch scoring (e.g., m5. large) to fit $50 credits.

Batch or real time
Batch only (monthly/on-demand). This avoids always-on endpoint costs and matches the publication cadence.

Model Monitoring:

Model monitoring

- anomaly rate drift
- score distribution drift
- reason-code distribution drift

Infrastructure monitoring

- job failure alarms
- runtime anomalies
- S3 input/output completeness checks

Data monitoring

- schema drift
- Missingness drift
- feature distribution drift (amounts, category mix, payer diversity)

Model CI/CD:

Checkpoints

- lint + unit tests
- schema tests
- pipeline integration test on sampled data
- train + evaluate gate
- register model + approval
- batch scoring job post-approval

Tests

- schema validation
- feature quality checks (ranges/missingness)
- evaluation gates (stability + anomaly rate bounds)
- security checks (IAM least privilege, S3 encryption)

Security Checklist, Privacy and Other Risks:

- **PHI:** No

- **PII:** Yes (provider identity/location). Justification: comes from public dataset; mitigation: encrypt storage, restrict access, do not use names as features, and present results as “review prioritization,” not wrongdoing claims.
- **User behavior tracked:** No
- **Credit card info:** No
- **S3 buckets:** raw/curated/features/preds (as listed)
- **Bias considerations:** differing payment patterns across specialties/regions may be legitimate; use peer-group comparisons and subgroup monitoring.
- **Ethical concerns:** outputs can be misused or misinterpreted; align wording and documentation with CMS guidance on interpretation.

Future Enhancements:

1. Add multi-level scoring (recipient-month + company-month + specialty-state benchmarks).
2. Add semi-supervised learning from reviewer feedback (“expected/unexpected”) to improve precision.
3. Improve explanations (e.g., SHAP on a supervised model trained from pseudo-labels).
4. Extend to Research Payments and Ownership/Investment datasets (separate pipelines).
5. Add automated data quality rules (missingness anomalies, schema changes across program years).

References

Centers for Medicare & Medicaid Services. (2025a). *Open Payments: Program overview and data updates (Program Year 2024 publication)*. Open Payments. https://openpaymentsdata.cms.gov/datasets/download

Centers for Medicare & Medicaid Services. (2025b). *Open Payments data dictionary / methodology documentation for public use files*. Open Payments. https://openpaymentsdata.cms.gov/dataset/e6b17c6a-2534-4207-a4a1-6746a14911ff#data-dictionary