## **Project Report - Medical Claims Fraud Detection: Anomaly detection in insurance claims**

**Introduction**

Fraudulent medical insurance claims pose a serious challenge to the financial sustainability of healthcare systems globally. These fraudulent activities can range from exaggerated billing and unnecessary procedures to completely fabricated claims. As insurance firms grow and handle increasing claim volumes, the risk of undetected fraud also increases, leading to billions of rupees in losses annually. Beyond the financial burden, these fraudulent actions erode trust in the healthcare system and can result in compromised patient care.

Insurance fraud detection is no longer just a manual audit process—it now requires intelligent systems that can analyse large volumes of data and spot irregularities instantly. Detecting anomalies or suspicious patterns in claim submissions is crucial to protecting the interests of both insurers and genuine policyholders.

In this project, we employ a **rule-based approach using Python** to flag potential fraud in a simulated medical claims dataset. Using domain expertise and known risk factors, we define rules that segment claims into different risk categories. We then use data visualizations to uncover trends and identify entities that warrant further investigation.

**Objective**

* Identify potentially fraudulent (high-risk) claims using clear business rules.
* Categorize claims into different risk levels.
* Rank providers based on the volume of high-risk claims.
* Visualize trends in claim risk levels over time.
* Provide actionable insights to assist in audit and fraud detection.

**Dataset Summary**

The dataset used contains **2000 simulated medical insurance claims**, mimicking real-world healthcare provider data. Each entry represents an individual claim submitted by a healthcare provider for reimbursement. The dataset includes the following variables:

| **Column Name** | **Description** |
| --- | --- |
| Patient ID | Unique alphanumeric identifier for each patient |
| Claim Amount | Total amount (in INR) billed for the claim |
| Claim Date | Date the claim was filed (YYYY-MM-DD format) |
| Provider ID | Unique alphanumeric code assigned to each healthcare provider |
| Procedure Code | Medical procedure code (standardized) billed in the claim |
| Procedure Count | Number of procedures billed under a single claim |
| Diagnosis Code | Primary ICD code representing the diagnosis |
| Length of Stay | Number of inpatient days for the patient (0 for outpatient claims) |
| Payment Mode | Method used for payment: Cash, Card, Insurance, etc. |

**Methodology**

1. **Data Generation**:

* A dummy dataset of 2000 rows was created using Python and NumPy.
* Values were randomly generated to reflect realistic claim behaviours, with appropriate ranges for claim amounts, procedures, and diagnosis codes.

1. **Rule-Based Risk Classification**:

* Business logic was used to assign risk levels:
  + - **High Risk**: Claim Amount > ₹100,000 **AND** Procedure Count > 6
    - **Medium Risk**: Claim Amount > ₹50,000 **OR** Procedure Count > 4
    - **Low/Normal Risk**: Remaining claims
* A new column Risk Level was created to store this classification.
* This approach mimics real-world claims auditing techniques where red flags trigger manual review.

1. **Provider Risk Summary**:

* The number of high-risk claims per provider was calculated.
* A bar chart visualization highlighted the top 10 providers with the highest risk ratios.
* This helps insurers identify patterns tied to specific facilities or doctors.

1. **Temporal Analysis**:

* Claims were grouped by month to observe the frequency of high-risk claims.
* Any spike in monthly numbers can be an indicator of seasonal or coordinated fraud activity.

**Results & Visuals**

**1. Top 10 Providers with Most High-Risk Claims**

The chart below shows the 10 providers with the highest count of high-risk claims.

A graph of red shades

AI-generated content may be incorrect.

**Interpretation:**

* Providers **P018**, **P012**, and **P008** show unusually high numbers of risky claims.
* These anomalies could be due to:
  + Inflated billing practices
  + Fake procedures added to claims
  + Repeat fraud behavior
* These providers should be subjected to deeper scrutiny, such as claim audits or investigations.

**2. Monthly Trend of High-Risk Claims**

The time series graph plots the number of high-risk claims per month.

A graph with blue lines and white text

AI-generated content may be incorrect.

**Observations:**

* Spikes in **June** and **December 2024** are especially concerning.
* These spikes may indicate fraudulent behavior to meet fiscal targets.
* Such patterns also suggest the need for enhanced detection during quarter-end or year-end.
* Trend graphs allow insurers to predict and pre-empt future spikes through monitoring systems.

**Implementation Details**

**Tools Used**

| **Tool** | **Purpose** |
| --- | --- |
| Python | Data handling, rule-based logic, risk modeling |
| Pandas | Dataset cleaning, transformation, aggregation |
| NumPy | Random data generation and numerical ops |
| Seaborn | Bar and trend chart creation |
| Google Colab | Notebook-based development and visualization |

**Code Workflow**

1. Import libraries
2. Generate and preprocess dataset
3. Apply risk classification rules
4. Group and analyze data by provider and time
5. Visualize top providers and monthly trends

This modular coding approach ensures flexibility, easy updates, and future scalability.

**Evaluation & Challenges**

**Effectiveness of Rule-Based Approach**

* **Advantages:**
  + Simple to implement and interpret.
  + High transparency in classification.
  + Easy to adjust rules based on expert input.
* **Limitations:**
  + May miss complex fraud patterns that don’t meet thresholds.
  + False positives if threshold values are poorly chosen.
  + Does not adapt over time like machine learning models.

**Challenges Faced**

* Balancing rule thresholds to avoid over/under-detection.
* Generating synthetic yet meaningful data patterns.
* Ensuring time-based trends were realistic for visual analysis.

**Business Relevance & Recommendations**

**Real-World Application**

* Helps insurance firms identify repeat offender providers.
* Supports audit teams with visual red flags.
* Can be scaled for real-time fraud alerts.

**Recommendations**

1. Integrate rule-based checks into claim processing systems.
2. Establish fraud scoring dashboards using Power BI.
3. Train claim reviewers on key indicators based on this model.
4. Expand risk analysis by integrating demographic and treatment history.

**Conclusion & Future Scope**

**Summary of Insights**

* Rule-based logic successfully segmented claims based on potential risk.
* Provider and monthly trend analysis provided clarity on fraud behavior.
* Visualization made the results more actionable for business teams.

**Future Enhancements**

* Integrate anomaly detection models (e.g., Isolation Forest).
* Use natural language processing (NLP) to parse claim descriptions.
* Include text-based notes and medical justifications for deeper audits.
* Add geolocation data to track regional fraud patterns.
* Explore real hospital data (with anonymization) to test and validate the model.