Unsupervised Anomaly Detection: Preprocessing and Modeling Plan

Introduction

This document outlines a detailed plan for preprocessing and modeling tasks in an unsupervised anomaly detection project. Without labeled data, the focus is on preparing the dataset and implementing unsupervised learning models to identify anomalies effectively.

1. Preprocessing Plan

1.1. 1. Data Cleaning and Transformation

Objective: Ensure the dataset is clean, consistent, and suitable for anomaly detection.

Tasks:

1. Handle missing values:
   - For numerical columns: Impute using median or k-nearest neighbors (KNN) imputation.
   - For categorical columns: Impute using the mode or ”unknown” placeholder.
2. Remove redundant features:
   - Drop columns with excessive missing data (¿50%).
   - Eliminate non-informative fields like unique identifiers.
3. Standardize formats for categorical and numerical data.

Deliverable: A cleaned dataset ready for encoding and scaling.

1.2. 2. Encoding Categorical Variables

Objective: Transform categorical features into numerical representations.

Tasks:

1. Use Label Encoding for low-cardinality features (e.g., Gender, Place of Service).
2. Use Frequency Encoding for high-cardinality features (e.g., Provider Type, HCPCS Code).

Deliverable: Encoded categorical columns in the dataset.

1.3. 3. Feature Scaling and Normalization

Objective: Standardize numerical data for consistent feature importance.

Tasks:

1. Apply StandardScaler for numerical columns like Medicare Payment Amount and Number of Services.
2. Use log transformation for skewed features (e.g., Charge-to-Payment Ratio).

Deliverable: Scaled numerical data for uniform distributions.

1.4. 4. Outlier Detection and Handling

Objective: Address extreme values that could distort unsupervised models.

Tasks:

1. Identify outliers using z-scores or the IQR method.
2. Cap extreme values to the 95th percentile or retain for anomaly detection.

Deliverable: Dataset with manageable outliers.

1.5. 5. Dimensionality Reduction

Objective: Simplify the dataset while retaining essential patterns.

Tasks:

1. Apply Principal Component Analysis (PCA) to reduce dimensions while retaining ¿95% variance.
2. Drop highly correlated features to avoid redundancy.

Deliverable: Reduced and optimized feature set.

1.6. 6. Data Splitting

Objective: Partition data for testing and validation.

Tasks:

1. Split the dataset into:
   - Training Set: 80% of the data for model fitting.
   - Validation/Test Set: 20% of the data for performance evaluation.
2. Use stratified sampling to maintain diversity across critical features.

Deliverable: Training and validation datasets.

2. Modeling Plan

2.1. 1. Exploratory Anomaly Detection

Objective: Experiment with multiple unsupervised models for anomaly detection.

Tasks:

1. Implement models such as:
   - Isolation Forest.
   - One-Class SVM.
   - DBSCAN (Density-Based Spatial Clustering).
2. Apply dimensionality reduction (e.g., PCA) for optimization.

Deliverable: Initial anomaly detection models with exploratory results.

2.2. 2. Clustering-Based Anomaly Detection

Objective: Separate anomalies from the main population using clustering algorithms.

Tasks:

1. Apply K-Means Clustering:
   - Calculate distances from cluster centroids to define anomalies.
2. Use Hierarchical Clustering to identify sparse or unusual clusters.
3. Apply DBSCAN to detect noise points as anomalies.

Deliverable: Cluster-based anomaly detection results.

2.3. 3. Deep Learning with Autoencoders

Objective: Leverage reconstruction errors to identify anomalies.

Tasks:

1. Build an autoencoder architecture with:
   - Input layer matching the feature dimensions.
   - Hidden layers for compression.
   - Output layer for reconstruction.
2. Train the model to minimize reconstruction loss.
3. Define anomaly thresholds based on reconstruction error.

Deliverable: Trained autoencoder model and defined anomaly thresholds.

2.4. 4. Model Evaluation

Objective: Assess model performance for unsupervised anomaly detection.

Tasks:

1. Use metrics such as:
   - Precision@Top-k for anomaly ranking.
   - Silhouette Score for clustering models.
   - Reconstruction Error Distribution for autoencoders.

2. Compare model outputs with domain expert feedback or simulated labels.

Deliverable: Comprehensive evaluation report.

2.5. 5. Ensemble Approach

Objective: Combine multiple models for enhanced detection.

Tasks:

1. Blend results from models (e.g., Isolation Forest, Autoencoders) using weighted averages or voting.
2. Validate ensemble performance using the same evaluation metrics.

Deliverable: Final ensemble model with optimized detection accuracy.