Anomaly Detection Pipeline Documentation

# Overview

This project implements a robust, production-ready anomaly detection pipeline for weekly count data, supporting multiple (target\_type, insight\_type) pairs and keys. The pipeline is designed to handle different seasonality, trends, and abrupt changes in the data, and provides:  
- Model selection (Negative Binomial with Fourier terms)  
- Change point detection (ruptures)  
- Robust fallback  
- Diagnostics and visualization

# Data Structure

Input CSV: Must contain at least the following columns:  
- insight\_date\_time (datetime)  
- target\_type (categorical)  
- insight\_type (categorical)  
- target\_key (categorical)

# Pipeline Steps

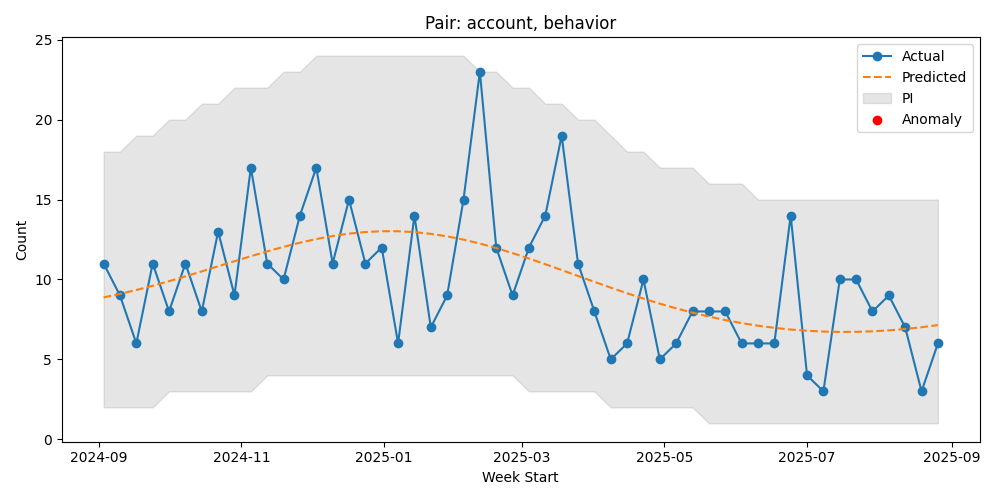
1. Data Aggregation  
 - Aggregates raw data to weekly counts for each (target\_type, insight\_type, target\_key).  
 - Fills missing weeks with zero counts.  
2. Model Fitting & Selection  
 - For each (target\_type, insight\_type) pair and key:  
 - Fits Negative Binomial (NB2) regression models with different numbers of Fourier terms (seasonality).  
 - Selects the best model using AIC.  
 - If model fitting fails, uses a robust rolling window fallback.  
3. Change Point Detection  
 - Uses the ruptures library (Pelt algorithm, model='l2', log1p transform) to detect change points in each time series.  
 - If a change point is found, splits the series and fits separate models to each segment.  
4. Anomaly Detection  
 - Calculates prediction intervals using the NB2 model.  
 - Flags anomalies where observed counts fall outside the prediction interval.  
5. Diagnostics & Visualization  
 - Outputs diagnostics (AIC, fallback usage, change points) for each segment to a CSV.  
 - Generates plots for each series, showing:  
 - Actual counts  
 - Predicted mean  
 - Prediction intervals  
 - Anomalies (red dots)  
 - Change points (purple vertical lines)  
 - Plots are saved in diagnostics\_plots/.

# Output Files

- light\_nb\_flags\_by\_key\_weekly.csv: Anomaly flags for each key.  
- light\_nb\_flags\_by\_pair\_weekly.csv: Anomaly flags for each pair.  
- diagnostics.csv: Model diagnostics for each segment.  
- diagnostics\_plots/: Plots for each pair and key.

# Example Visualization

Below is an actual example plot generated by the pipeline:



Explanation:  
- Blue line: Actual weekly counts  
- Dashed line: Model-predicted mean  
- Gray band: Prediction interval  
- Red dots: Detected anomalies  
- Purple lines: Detected change points

# Code Structure

lightweight\_nb\_anomaly.py: Main script implementing the pipeline.  
 - Key functions: ensure\_weekly\_counts, build\_design, fit\_nb2, predict\_pi, robust\_fallback, detect\_change\_points, plot\_series, main  
run\_lightweight.py: Runner script to execute the pipeline with the correct input/output paths.

# Requirements

Python 3.7+  
pandas  
numpy  
statsmodels  
ruptures  
matplotlib  
Install with:  
 pip install pandas numpy statsmodels ruptures matplotlib

# Usage

1. Place your input CSV in the correct location.  
2. Update run\_lightweight.py if needed.  
3. Run:  
 python run\_lightweight.py  
4. Review outputs in the out\_lightweight directory.

# Notes

- The pipeline is robust to missing data, overdispersion, and abrupt changes.  
- Diagnostics and plots help validate model quality and anomaly detection.  
- You can tune the change point penalty (pen in detect\_change\_points) and model complexity (fourier\_K) as needed.

# Contact

For further customization or support, contact your data science team.

# Gen AI Prompt for Copilot or ChatGPT

If you want to re-create this pipeline on any new data, use this prompt:  
  
I have weekly count data in CSV format, with columns like `target\_type`, `insight\_type`, `target\_key`, `insight\_date\_time`, and I want to detect anomalies for each (target\_type, insight\_type) pair and key.  
  
Please develop a robust Python pipeline that:  
- Aggregates the data to weekly counts for each (target\_type, insight\_type, target\_key).  
- For each (target\_type, insight\_type) pair and key:  
 - Fits multiple Negative Binomial (NB2) models with different numbers of Fourier terms (for seasonality) and selects the best model using AIC.  
 - Detects change points in the time series using the `ruptures` library (with a log1p transform and `model='l2'`), and if a change point is found, fits separate models to each segment.  
 - Uses a robust rolling window fallback if model fitting fails.  
 - Calculates prediction intervals using the NB2 model and flags anomalies outside the interval.  
 - Collects diagnostics (AIC, fallback usage, change points) for each segment and saves them to a CSV.  
 - Generates and saves plots for each series, showing actuals, predictions, intervals, anomalies, and change points.  
- Outputs:  
 - Anomaly flags for each key and pair as CSVs.  
 - Diagnostics CSV.  
 - Plots for each series in a diagnostics folder.  
- Use only open-source Python libraries (pandas, numpy, statsmodels, ruptures, matplotlib).  
- The code should be robust, modular, and ready to run on new data with similar structure.  
  
Please provide the full code, including requirements, and ensure it is thoroughly tested and handles edge cases.