Title: Anomaly Detection Workflow with Python
Subtitle: A Step-by-Step Guide Using Machine Learning and Automation Tools
Date: March 24, 2025

1. Importing Libraries
- pandas: Data manipulation
- numpy: Numerical operations
- sklearn.ensemble.IsolationForest: Anomaly detection
- sklearn.preprocessing: Encoding and scaling
- smtplib, email.mime: Email handling
- logging: Debugging
- requests, openai, jira: API interactions
- os, sys: System operations

2. Logging Setup
- logging.basicConfig(level=logging.INFO): Tracks INFO-level messages

3. API and Configuration Setup
- OpenAI: openai.api_key = "your-openai-api-key"
- Jira: jira_url, jira_user, jira_token
- Agentic AI: AGENTIC_API_KEY = "your-agentic-api-key"

4. Data Loading and Preprocessing
- Checks file existence, loads CSV, validates columns
- Label encoding for categorical data, date conversion

5. Feature Engineering
- Cleans numeric data, sorts/groups, creates features (e.g., Days_Since_Last, GL_Change)

6. Contamination Estimation
- Estimates outliers using Z-score, sets contamination level

7. Anomaly Detection (Isolation Forest)
- Scales data, trains model, labels anomalies

8. Generating Anomaly Explanations Using OpenAI
- Creates prompt, uses OpenAI to explain anomalies

9. Creating Tasks in Agentic AI and JIRA
- Agentic AI: POST task via API
- Jira: Creates ticket with anomaly details

10. Sending Email Notification
- Sends HTML-formatted email via SMTP

11. Feedback Loop and Model Retraining
- Adjusts model based on feedback, retrains

12. Main Execution
- Loads data, trains model, sends email

13. Execution Example
- file_path = "/path/to/your/file.csv"
- main(file_path, feedback_file_path)