Documentation for Anomaly Detection in Bank Transactions

---

1. Defining the Question

a) Specifying the Question

The primary goal is to identify anomalies within bank statement transactions.

b) Defining the Metric for Success

The success metrics are defined as follows:

1. Achieving a model accuracy of at least 70%.
2. Attaining a model precision score greater than 0.70 for both presence and absence on the training dataset.

c) Understanding the Context

Bank statements are crucial for analyzing an individual's or a company's financial transactions. However, different formatting across banks makes this analysis time-consuming and labor-intensive. With the increased governmental focus on curbing Money Laundering, Terrorism Financing, and Tax Evasion, there's a need for a model to detect transaction anomalies to aid security agencies in identifying malicious financial activities.

d) Recording the Experimental Design

The Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology was employed in this project.

---

2. Reading the Data

In this step, necessary libraries are imported and the dataset is loaded into a Pandas DataFrame from a CSV file.

---

3. Checking the Data

Initial data exploration is conducted to understand the dataset's structure, check for missing values, and examine basic statistics.

---

4. Data Cleaning

Here, irrelevant features are dropped, the date column is converted to a datetime object, and any missing values are dealt with accordingly.

---

5. Exploratory Data Analysis (EDA)

Various statistical and visual analyses are performed to understand the data's distribution and trends.

Feature Engineering

Two new features are engineered:

1. `sum_5days`: Cumulative withdrawal amounts from an account over the previous 5 days.
2. `count_5days`: Count of withdrawal transactions from an account over the previous 5 days.

These features aim to assist in detecting large transactions done below the reporting threshold over a period of time.

Anomaly Detection: Isolation Forest

Installation and Importing Libraries

The PyOD library is installed, and necessary libraries are imported for anomaly detection using the Isolation Forest algorithm.

Model Training and Evaluation

1. The `IForest` model from PyOD is initialized with a specified contamination proportion to indicate the percentage of outliers in the dataset.
2. The model is trained on the engineered features, and predictions along with anomaly scores are obtained.
3. A visualization is created to demonstrate the decision boundary of the Isolation Forest, distinguishing between inliers and outliers based on the newly created features.

In this visualization, the red contour line represents the decision function's threshold separating inliers (white dots) from outliers (black dots), where inliers are normal transactions and outliers are anomalous transactions.

The x-axis represents the count of withdrawal transactions over 5 days, and the y-axis represents the sum of withdrawal transactions over 5 days. Through this visual representation, it's easier to comprehend the regions where anomalies are prevalent, aiding in better understanding and interpretation of the results obtained from the Isolation Forest anomaly detection model.

This documentation provides an overview of the process undertaken to develop a model for identifying anomalous transactions in bank statements. Each section is broken down into sub-sections for better understanding and readability.