Visualization of Anomalies: Summary Report

Introduction and Objectives

Objective

The objective of this task is to create, refine, and present visualizations that allow for a clearer understanding of anomaly detection patterns and model performance. Specifically, we aim to identify and analyze false positives and false negatives generated by two models: Isolation Forest and Local Outlier Factor (LOF).

Approach

1. Develop interactive time-series visualizations for Isolation Forest and LOF anomaly scores over time.
2. Highlight false positives and false negatives through scatter plots and categorize anomalies by model type.
3. Incorporate interactive elements such as toggles and sliders to improve user experience.
4. Refine visualization strategies based on feedback to enhance insight and clarity.

Step-by-Step Process

Step 1: Importing Libraries

We start by importing the necessary libraries, including `pandas` for data manipulation and `plotly` for creating interactive visualizations.

```
python
```

```
Copy code
import plotly.graph_objects as go
import pandas as pd
```

Step 2: Isolation Forest Anomaly Scores Over Time

An interactive time-series plot was created for Isolation Forest scores. This plot provides an initial view of anomaly scores over time, using sliders to allow users to zoom in on specific time intervals.

```
python
Copy code
fig = go.Figure()
fig.add_trace(go.Scatter(
    x=data['Time'],
    y=data['iso_forest_score'],
    mode='lines',
    name='Isolation Forest Score',
    line=dict(color='blue')
))
```

Step 3: Adding LOF Scores and Toggle Option

A line plot for the LOF anomaly scores was added, along with a toggle option to switch between Isolation Forest and LOF scores. This feature helps users focus on individual model performances.

```
python
Copy code
fig.add_trace(go.Scatter(
    x=data['Time'],
    y=data['lof_score'],
    mode='lines',
```

```
    name='LOF Score',
    line=dict(color='orange')
))
```

Interactive Visualization Enhancements

Step 4: Toggle and Slider for Interactive Viewing

To further enhance interactivity, we added a dropdown toggle and slider for users to view each model's scores individually or together, as well as a zoom feature for detailed examination.

```
python
Copy code
fig.update_layout(
    updatemenus=[
        {
            'buttons': [
                {'args': [{'visible': [True, False]}], 'label': 'Isolation Forest Only', 'method': 'update'},
                {'args': [{'visible': [False, True]}], 'label': 'LOF Only', 'method': 'update'},
                {'args': [{'visible': [True, True]}], 'label': 'Both', 'method': 'update'}
            ],
            'direction': 'down',
            'showactive': True
        }
    ],
```

```
    sliders=[{
        'active': 0,
        'currentvalue': {'prefix': 'Time Range: '},
        'steps': [
            {'label': 'Full Range', 'method':
'relayout', 'args': [{'xaxis.range':
[data['Time'].min(), data['Time'].max()]}]},
            {'label': 'Zoom In', 'method':
'relayout', 'args': [{'xaxis.range':
[data['Time'].iloc[0], data['Time'].iloc[100]]}]}
        ]
    }]
)
```

Visualizing False Positives and False Negatives

Step 1: Define False Positives and False Negatives

False positives and false negatives were defined based on model predictions and actual anomaly labels. These visualizations allow us to evaluate each model's performance in distinguishing normal and anomalous data points.

```
python
Copy code
# Isolation Forest
data['iso_false_positive'] =
((data['iso_forest_pred'] == 1) &
(data['true_anomaly'] == 0)).astype(int)
```

```
data['iso_false_negative'] =
((data['iso_forest_pred'] == -1) &
(data['true_anomaly'] == 1)).astype(int)
```

Step 2: Plotting Isolation Forest False Positives and Negatives

Separate scatter plots were created to display Isolation Forest false positives and negatives, providing insight into areas where the model incorrectly identified anomalies.

```
python
Copy code
fig_fp_iso =
px.scatter(data[data['iso_false_positive'] == 1],
x='PCA1', y='PCA2',
                        title='Isolation Forest False
Positives',

color_discrete_sequence=['red'])
fig_fn_iso =
px.scatter(data[data['iso_false_negative'] == 1],
x='PCA1', y='PCA2',
                        title='Isolation Forest False
Negatives',

color_discrete_sequence=['blue'])
```

LOF Model Visualizations and Combined Plots

Step 3: LOF Model False Positives and False Negatives

Similar scatter plots were created for the LOF model to visualize its false positives and negatives, enabling comparative analysis with the Isolation Forest model.

```
python
Copy code
fig_fp_lof =
px.scatter(data[data['lof_false_positive'] == 1],
x='PCA1', y='PCA2',
                        title='LOF False Positives',

color_discrete_sequence=['orange'])
fig_fn_lof =
px.scatter(data[data['lof_false_negative'] == 1],
x='PCA1', y='PCA2',
                        title='LOF False Negatives',

color_discrete_sequence=['purple'])
```

Step 4: Combined Visualization of Anomaly Types

To provide a comprehensive view, a single plot was created to show all anomalies (false positives, false negatives) by model type. This plot makes it easier to identify patterns across both models.

```
python
Copy code
data['anomaly_type'] = data.apply(lambda row:
    'ISO False Positive' if row['iso_false_positive']
== 1 else (
```

```
    'ISO False Negative' if row['iso_false_negative']
== 1 else (
    'LOF False Positive' if row['lof_false_positive']
== 1 else (
    'LOF False Negative' if row['lof_false_negative']
== 1 else 'Correct'))), axis=1)

fig_combined = px.scatter(data, x='PCA1', y='PCA2',
color='anomaly_type',
                          title='False Positives and
False Negatives by Anomaly Type and Model')
```

Insights and Feedback-Driven Refinements

Insights

- **Model Sensitivity**: Isolation Forest and LOF models display different sensitivity levels, with one being more prone to false positives in certain data clusters.
- **Anomaly Patterns**: Patterns, such as clusters in false positives and negatives, help identify specific areas where models struggle, potentially indicating common features in misclassified data points.
- **False Positives and Negatives**: Isolation Forest and LOF differ in their false positive and false negative distribution, highlighting unique strengths and weaknesses in anomaly detection for each.

Feedback-Driven Refinements

- **Improved Labeling**: Enhanced readability by increasing font sizes and adding more descriptive axis titles.
- **Color Adjustments**: Colors were modified to provide better distinction between anomaly types and models.
- **Annotations**: Added key data point annotations in time-series plots to mark significant anomalies, spikes, or trends.

Conclusion and Recommendations

Summary of Findings

The refined visualizations provide a comprehensive view of model behavior, allowing for a better understanding of false positives, false negatives, and anomaly patterns over time. By using toggles and interactive elements, the visualizations allow for detailed, flexible exploration of model performance.

Recommendations

1. **Model Tuning**: Based on observed false positive and negative patterns, further parameter tuning for Isolation Forest and LOF is recommended.
2. **Additional Metrics**: Future analysis could include precision-recall scores per anomaly type to evaluate model accuracy quantitatively.
3. **Visualization Refinements**: Continue to refine visual clarity and interactivity based on additional feedback, particularly from end-users who would utilize these visualizations in a production environment.

This report provides a structured view of anomaly detection patterns and highlights potential improvements for both modeling and visualization strategies.