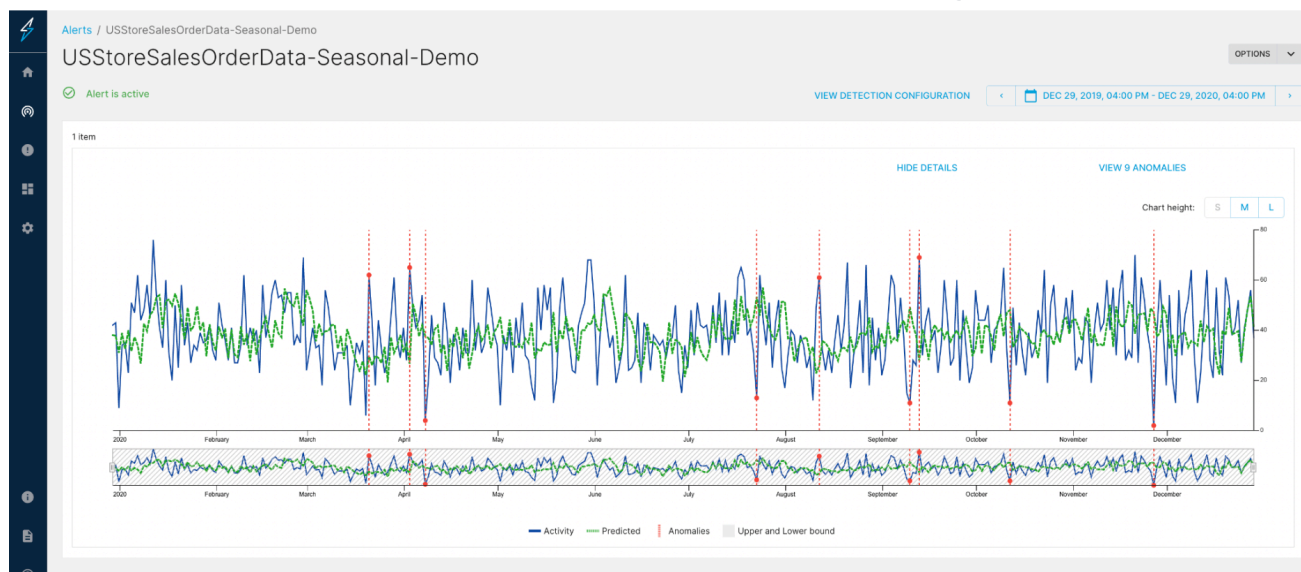


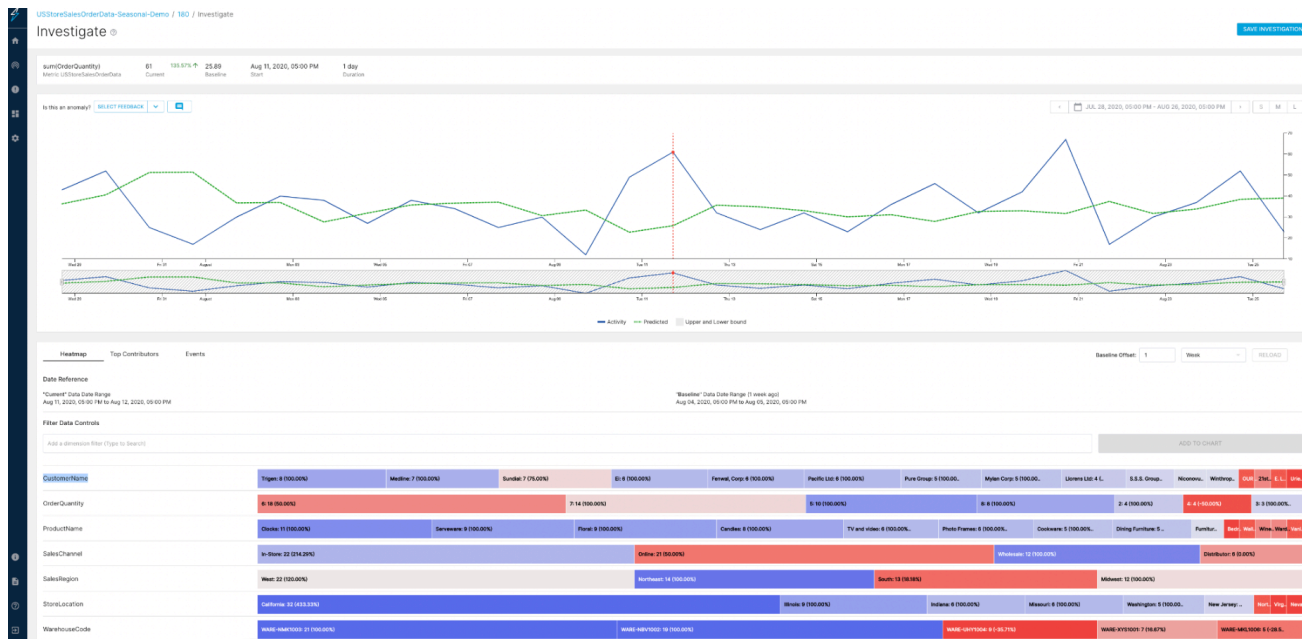
△ > StarTree ... > How-To ... > ThirdEye ... > Monitoring Store Sales KPI for an eCommerce retail store

Monitoring Store Sales KPI for an eCommerce retail store

In this recipe, we'll use Startree ThirdEye to monitor the *number of orders across multiple dimensions such as products, sales channel, sales region, and store locations* using StarTree ThirdEye.

Number of orders is a critical metric to measure the success of the business performance of eCommerce retail stores. In addition to monitoring the orders, this recipe covers how to perform root-cause analysis to identify which dimensions changed and contributed the most to the outlier/deviation in the metric (spike or drop, or trend change).





Common challenges faced by eCommerce retail stores in sales

The COVID-19 pandemic disrupted e-commerce retail stores globally. Since 2020 there has been a significant surge in online shopping, and eCommerce retail stores are struggling to keep up with the demand. Retailers face challenges such as managing demand vs supply, pricing, and trend prediction.

- Unexpected surges in-store sales can be a result of holiday sales or due to a popular campaign run or pricing changes. In ability to spot the right reason in right time can result in lost opportunities to grow the business.
- An unexpected drop in-store sales can be a result of competition or user behavioral changes or market conditions or change in pricing etc. In ability to find the root-cause in a timely fashion can result in revenue losses.

What is the solution? Anomaly detection in store sales

As online sales surge, eCommerce retail stores generate a huge amount of data, making it difficult to make decisions. Anomaly detection can help you identify fluctuation in store sales,

which lets you mitigate risk and open up avenues for higher sales and performance. To drive better business performance and customer engagement, you must constantly monitor the volume of sales, related dimensions, and detect anomalies across your business. ThirdEye helps detect an anomaly, notify operations and product analytics teams of the root cause, and supports making data-informed business decisions.

The following are a few examples of types of anomalies found in eCommerce retail store sales:

- **Order sales anomaly (sudden spike in sales):** For eCommerce retail companies, anomaly detection techniques assist teams with monitoring any sudden changes in the number of completed transactions, including an unexpected surge in demand. Spikes in demand occur for a number of reasons, including holiday shopping and sudden changes in weather conditions. By identifying increases in sales quickly, businesses can analyze their inventory levels and make important business decisions, such as ordering additional supplies or identifying alternative products that effectively meet their customers' needs.
- **Delivery anomaly:** distribution provides guidelines to ensure that products seamlessly move through the warehouses and is stocked for the right amount of time. Sometimes the distribution time does not follow the trend and results in anomalies.
- **Pricing anomaly:** Anomaly detection solution makes it easier to spot hard-to-find glitches, such as a mispriced item. If revenue dropped lower than expected but conversion rates remained high, then there's a chance that an item was mispriced, or that a teammate accidentally let a promotion go on for too long.
- **Monitoring marketing campaigns:** Retailers and e-commerce businesses can use anomaly detection technology to evaluate the effectiveness of a marketing campaign or new promotions.
- **Identifying business opportunities:** The anomaly detection solution assist with identifying new business opportunities, preventing revenue leakage, and better understanding of target customer's demographics, including age, gender, and geographic location. Developing a deeper understanding of user, and their buying behaviors, is critical to deliver the products, goods, and services that users want and need.

How StarTree ThirdEye can help improve the store sales of an eCommerce retail store?

Business metrics can be explored and analyzed in any Data Visualization, and reporting platform such as Superset, Tableau, and PowerBI. However, the visualization tools do not support:

- Analyzing multi-dimensional metrics
- Smart alerting mechanisms with shifting metrics trends
- RCAs (Root Cause Analysis)

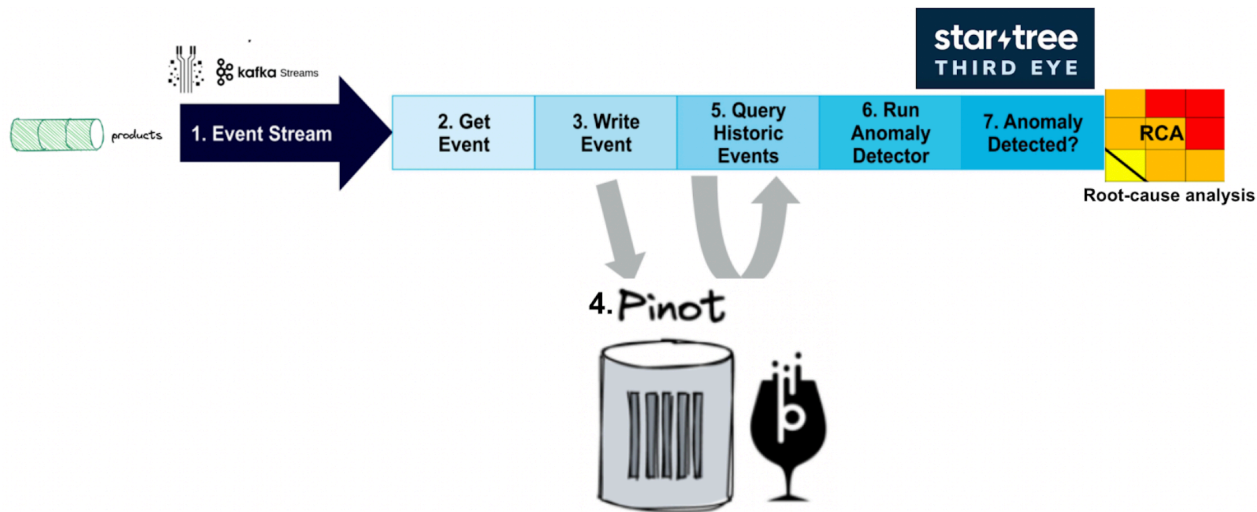
Using StarTree ThirdEye the following benefits can be achieved

- **Dimesion drills:** Determining which store locations and sales regions were impacted and by how much on a daily basis is a laborious task. StarTree ThirdEye can analyze multi-dimensional metrics (Eg: volume of order sales can be analyzed by multiple dimensions such as store locations, sales regions, and warehouses) and can generate multiple time series for the same metric. Eg: To monitor the “volume of order sales” metric across many sales regions and store locations, it will generate many time series’ for the “# of orders” metric across all the permutations of Sales regions and store locations
- **Robust smart detection and tuning:** Lacking a robust and smart alerting mechanism to identify the most impacted area will result in blind spots and increased costs due to manual investigations. StarTree ThirdEye has a smart alerting feature that can configure a multi-level alerting hierarchy based on the severity of the alerts and the multiple dimensions of a metric with a single alert configuration. It sends notifications to slack, email, and custom notifications platforms (using webhook). It can also associate with custom events (such as browser upgrades that could have resulted in a drop in page views).
- **Identifying the Root Cause Analysis** across multiple dimensions and related events in real-time is of the hardest challenges for every organization. StarTree ThirdEye can perform Root Cause Analysis across metrics, dimensions, and the impact of one event on the metrics in an easy-to-consumable graphic.

Interested in trying out StarTree ThirdEye?

Follow this [link](#) to find different deployment options for trying out StarTree ThirdEye.

Architecture diagram



Here are the steps to quickly try this “eCommerce retail store sales recipe and try StarTree ThirdEye anomaly detection” for real-time or near real-time monitoring

- Data preparation and availability
- Data ingestion
- Alert creation and notifications
- Tuning anomaly detection models (if needed based on use-case and context)
- Anomaly reporting
- Root cause analysis: Anomaly investigation

Data Preparation and Availability

- Identify key metrics to monitor. The core metrics for learning about store sales is:
 - Number of orders
 - Unit price
- Identify the dimensions that are applicable for a given metric:
 - SalesRegion, StoreLocation, WarehouseCode, SalesChannel

- Decide granularity for your detection:
 - Granularity enables you for real-time or near real-time monitoring (hourly/daily/weekly time series).

Types of time series data



Data Ingestion

You can download this CSV file [link](#) Ingest this CSV data to Pinot using Data Manager [link](#) or Pinot API.

The Pinot schema should look like this

```
{
  "schemaName": "USStoreSalesOrderData",
  "dimensionFieldSpecs": [
    {
      "name": "SalesChannel",
      "dataType": "STRING"
    },
    {
      "name": "WarehouseCode",
      "dataType": "STRING"
    }
  ]
}
```

```
    },
    {
      "name": "SalesRegion",
      "dataType": "STRING"
    },
    {
      "name": "CustomerName",
      "dataType": "STRING"
    },
    {
      "name": "StoreLocation",
      "dataType": "STRING"
    },
    {
      "name": "ProductName",
      "dataType": "STRING"
    },
    {
      "name": "OrderQuantity",
      "dataType": "LONG"
    }
  ],
  "metricFieldSpecs": [
    {
      "name": "DiscountApplied",
      "dataType": "DOUBLE"
    },
    {
      "name": "UnitPrice",
      "dataType": "DOUBLE"
    },
    {
      "name": "UnitCost",
      "dataType": "DOUBLE"
    }
  ],
  "dateTimeFieldSpecs": [
    {
      "name": "OrderDate",
      "dataType": "STRING",
      "format": "1:MILLISECONDS:SIMPLE_DATE_FORMAT:yyyy-MM-dd",
      "granularity": "1:MILLISECONDS"
    },
    {
      "name": "ShipDate",
      "dataType": "STRING",
      "format": "1:MILLISECONDS:SIMPLE_DATE_FORMAT:yyyy-MM-dd",
      "granularity": "1:MILLISECONDS"
    },
  ],
```

```

{
  "name": "DeliveryDate",
  "dataType": "STRING",
  "format": "1:MILLISECONDS:SIMPLE_DATE_FORMAT:yyyy-MM-dd",
  "granularity": "1:MILLISECONDS"
}
]
}

```

Note: The “dateTimeFieldSpecs” will be used to set the granularity and understand the seas

Alert creation and notifications

- Refer to how to guide to create an alert [link](#) and use the following alert configurations to create alerts
- Subscribe to notifications [link](#)

Use the following alert configuration for “Simple threshold based alert creation”

```

{
  "name": "threshold-alert-order-sales-sample-alert",
  "description": "Threshold alert on order sales",
  "template": {
    "name": "startree-threshold"
  },
  "templateProperties": {
    "dataSource": "pinot",
    "dataset": "USOrderSalesData",
    "aggregationFunction": "avg",
    "aggregationColumn": "OrderQuantity",
    "monitoringGranularity": "P1D",
    "max": "20",
    "min": "4"
  },
  "cron": "0 0 0/2 1/1 * ? *"
}

```

Use the following alert configuration for “Startree-ets alert creation” (advanced detection model based on metrics pattern and seasonality)


```
{
  "name": "USSalesOrderData-anomaly-detection",
  "description": "Uses template startree-ets.",
  "template": {
    "name": "startree-ets"
  },
  "templateProperties": {
    "dataSource": "pinot",
    "dataset": "USOrderSalesData",
    "aggregationFunction": "sum",
    "aggregationColumn": "OrderQuantity",
    "seasonalityPeriod": "P7D",
    "lookback": "P120D",
    "monitoringGranularity": "P1D",
    "sensitivity": "3"
  },
  "cron": "0 0 0 1/1 * ? *"
}
```

Use the following alert configuration for “Startree-ets-dx alert creation” (advanced detection model based on metrics pattern and seasonality + dimension exploration (multiple timeseries)) (Learn more about the concepts by clicking this [link](#))

```
{
  "name": "USStoreSalesOrderData-Seasonal-DX-Demo",
  "description": "Uses template startree-ets-dx.",
  "template": {
    "name": "startree-ets-dx"
  },
  "templateProperties": {
    "dataSource": "pinot",
    "dataset": "USOrderSalesData",
    "aggregationFunction": "sum",
    "aggregationColumn": "OrderQuantity",
    "seasonalityPeriod": "P7D",
    "lookback": "P120D",
    "monitoringGranularity": "P1D",
    "sensitivity": "3",
    "rcaAggregationFunction": "sum",
    "queryFilters": "${queryFilters}",
    "enumerationItems": [
      {

```

```

    "name": "overall",
    "params": {
      "queryFilters": ""
    }
  },
  {
    "params": {
      "queryFilters": " AND SalesRegion = 'Midwest'"
    }
  },
  {
    "params": {
      "queryFilters": " AND SalesRegion = 'West'"
    }
  },
  {
    "params": {
      "queryFilters": " AND SalesRegion = 'West' AND SalesChannel = 'In-Store'"
    }
  },
  {
    "params": {
      "queryFilters": " AND SalesRegion = 'Northeast'"
    }
  }
]
},
"cron": "0 0 0 1/1 * ? *"
}

```

Tuning anomaly detection models

To reduce the number of false alarms, based on the business logic and contexts, StarTree ThirdEye allows you to customize anomaly criteria in the following way (Learn more about alert tuning by clicking this [link](#)):

Adjust the sensitivity and rules (you can use query filters) at various levels (metric, group, series):

- Metric level: In the alert config you have just provided metric and no filters then it will tune the time series under that metrics
- Specific dimension value or group of dimensions: In the alert, config specify query filters and this gives the ability to tune the time series at a group of dimensions level for the given

metrics

- Once anomaly detected feedback can be given on the anomaly or missed anomaly can be reported for adaptive learning and better accuracy
- StarTree ThirdEye allows you to configure "Anomaly filters"
 - coldstart filter
 - weekday filter
 - threshold filter
 - guardrail metric threshold filter
 - holiday filter

```
{
  "name": "USStoresSales-sample-tune-alert",
  "description": "US Store sales sample tune alert",
  "template": {
    "name": "startree-ets"
  },
  "templateProperties": {
    "dataSource": "pinot",
    "dataset": "USOrderSalesData",
    "aggregationFunction": "sum",
    "lookback": "P28D",
    "monitoringGranularity": "P1D",
    "sensitivity": "8",
    "aggregationColumn": "OrderQuantity",
    "queryFilters": " AND SalesRegion = 'West' AND SalesChannel = 'In-Store'",
    "eventSqlFilter": "'US' member of dimensionMap['countryCode']",
    "eventTypes": [
      "HOLIDAY"
    ],
    "daysOfWeek": [],
    "hoursOfDay": [],
    "dayHoursOfWeek": {},
    "thresholdFilterMin": "-1",
    "thresholdFilterMax": "-1",
    "guardrailMetricMin": "-1",
    "guardrailMetricMax": "-1",
    "guardrailMetric": "COUNT(*)"
  },
  "cron": "0 0 5 ? * * *"
}
```

Anomaly reporting

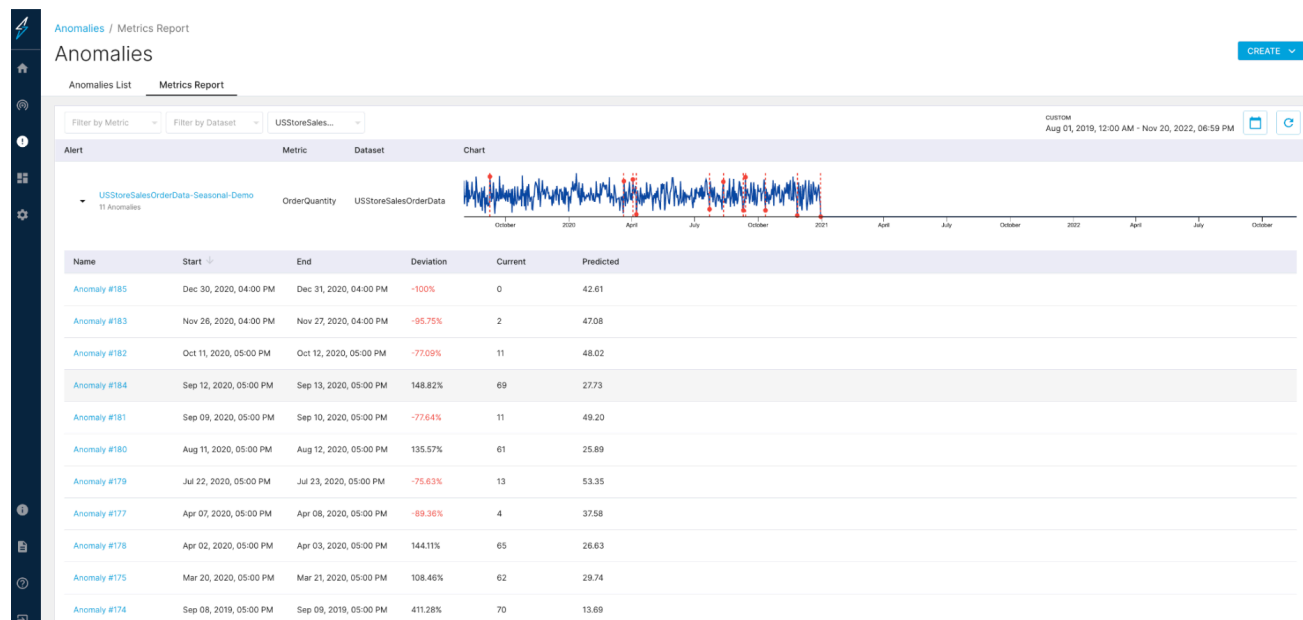
Anomalies can be reported in multiple ways using ThirdEye, such as:

- Email
- Slack
- Webhook
- Using APIs
- Viewing the anomalies directly in ThirdEye

Here is the [link to a “how to guide”](#) which describes how the different types of notifications (Email, Slack, Webhook) can be used to report anomalies.

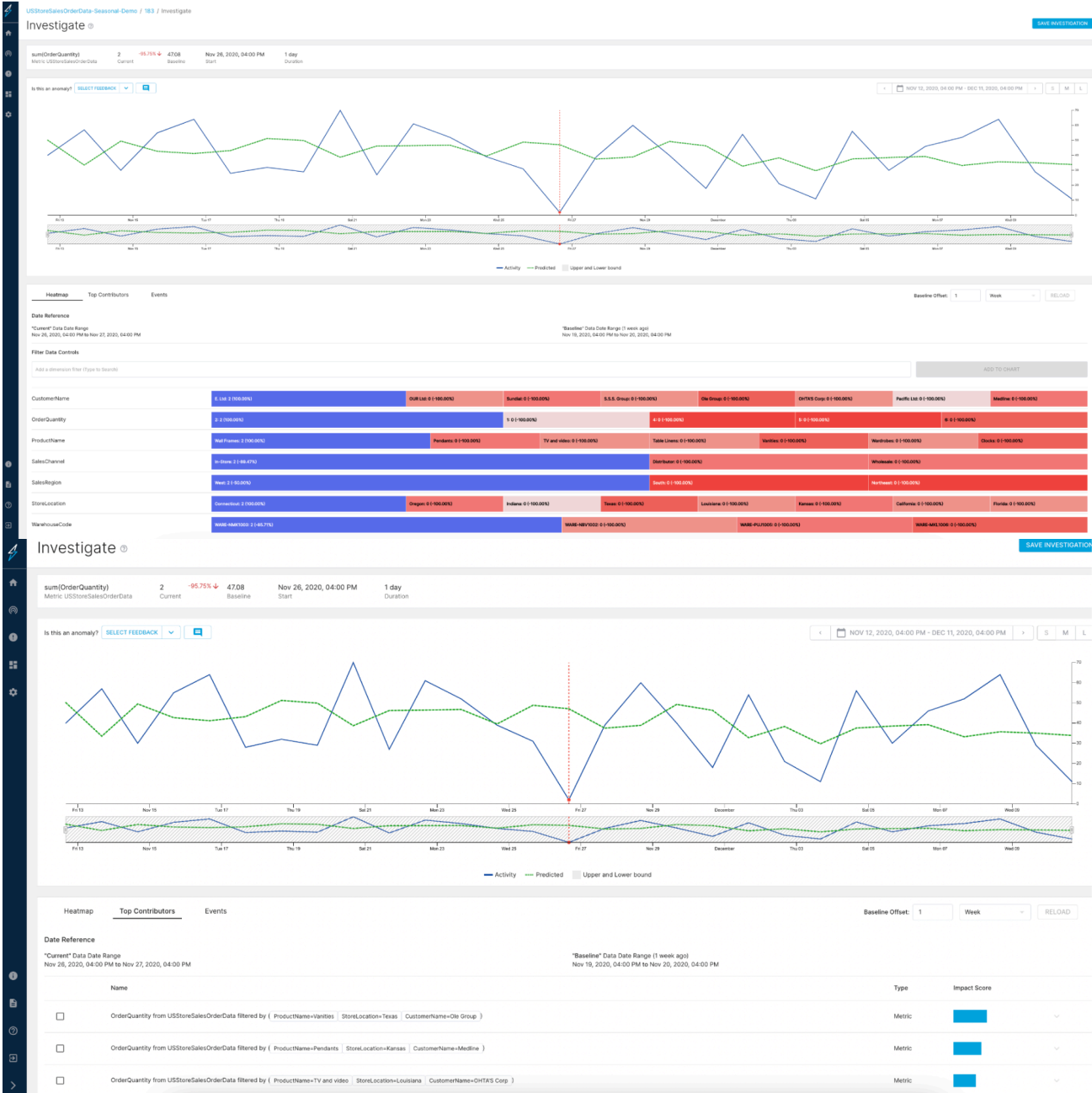
You can also pull these anomalies using the ThirdEye API [link to guide](#) and report those anomalies your custom application.

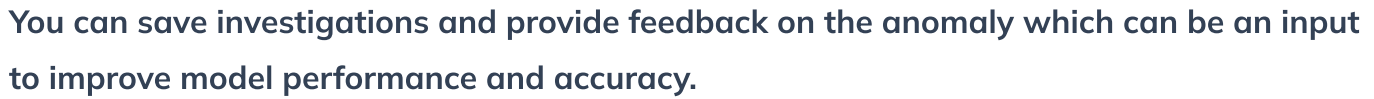
Access anomalies directly in the ThirdEye UI. [link](#)



Root-cause analysis: anomaly investigation

Here is the how-to guide to perform root cause analysis by investigating anomalies with signals from the heatmap, top contributors, and custom events: [link](#).





Last updated on May 21, 2024