

Splunk MLTK Hands-On Lab Guide

Exercise 1: Outlier Detection (Smart Outlier Detection)

Scenario Overview

Outlier Detection is about finding data points that *don't fit* the normal pattern. These unusual points are called **outliers** or anomalies.

In a supermarket context, an outlier could be a purchase that is **much higher or lower than typical** – for example, if most customers buy 2-3 items but one purchase has 100 items, that 100-item sale is an outlier. Detecting such anomalies matters because they can reveal important issues or insights. An outlier might indicate a **data error**, an instance of **fraud** (like an abnormal purchase pattern), or a one-time event affecting sales^[1]. By identifying outliers in retail purchase data, stores can investigate and respond to unusual situations (e.g. sudden spikes or drops in sales) before they become problems.

Step-by-Step Instructions

1. **Open the MLTK App and Navigate to Examples:** In your Splunk instance, launch the **Machine Learning Toolkit** app. Once inside MLTK, click on the **Examples** (Showcase) section. From the list of example categories, find **Outlier Detection** (sometimes labeled "Detect Outliers").
2. **Select the Supermarket Purchases Anomaly Example:** Under Outlier Detection examples, click on "**Find Anomalies in Supermarket Purchases.**" This example is a pre-built scenario that will help us detect outliers in supermarket sales data. (It uses a sample dataset of supermarket purchase quantities across different stores.)
3. **Create a New Smart Outlier Experiment:** After selecting the example, you will be prompted to create a new experiment using the Smart Outlier Detection assistant. In the Experiments view, choose **Smart Outlier Detection** as the experiment type, then click the **Create New Experiment** button (usually at the top right of the screen)^[2].
4. **Name the Experiment:** A dialog will appear to set up your experiment. Enter a name for your experiment (for example, "**Supermarket Purchase Outlier Detection**") and add a brief description if you like. This helps identify the experiment later. Once done, click **Create** to launch the Outlier Detection assistant interface^[3].
5. **Define the Data for Analysis:** In the **Define** stage of the assistant, select the dataset that contains the supermarket purchase data. The default **Dataset** "**Supermarket purchases (supermarket.csv)**" will be automatically populated – this is the sample dataset provided for the example^[4]. Select this dataset to load it

into the experiment. The data represents purchase transactions (including quantity of items purchased, store ID, etc.) over time. After selecting it, you can use the preview to confirm the data looks correct (optional).

6. **Proceed to the Learn Stage:** With the dataset selected, click **Next** (at the top right) to move to the **Learn** stage of the assistant[5]. In this stage, the Smart Outlier Detection assistant will automatically apply an outlier detection model to your data. (No coding is required – the assistant uses a density-based algorithm under the hood to model what “normal” purchases look like[6].) You might see options or settings, but for this beginner exercise the defaults are fine.
7. **Run the Outlier Detection:** The assistant will now train the model and identify anomalies. It may run automatically upon entering the Learn stage, or you might need to click a **Run** or **Detect Outliers** button (depending on your MLTK version). The toolkit will analyze the purchase quantities in the dataset and look for data points that deviate significantly from the typical range for each store (baseline behavior).
8. **Review the Results:** Next, click on the **Review** stage (or it may load automatically after learning). Here you will see the results of the outlier detection. The interface typically shows a visualization (such as a line chart or scatter plot) with normal data points and any **outliers highlighted**. For example, points that represent abnormally large purchase quantities may be marked in a different color or with an “outlier” label. You will also see a summary of how many outliers were found. Take a moment to identify those anomalies – these are the supermarket purchases that **deviate from the baseline** pattern.
9. **Interpret the Findings:** Look at the flagged outliers and consider why they might be unusual. Are they single transactions with an extremely high item count? Do they occur on certain dates or at specific stores? This step is about understanding what the anomalies mean. In a real scenario, a store manager might investigate whether a large outlier sale was due to a bulk purchase, a data entry mistake, or a promotional event. (For our lab data, it’s just for practice, but the concept carries over to real retail data.)
10. **Conclude the Outlier Exercise:** You have now successfully used the Smart Outlier Detection assistant to find anomalies in supermarket purchase data. Feel free to discuss or note down what the outliers were and why catching them is important. When ready, you can proceed to the next exercise.

Summary – Key Takeaways from Outlier Detection

- **What is an Outlier?** An outlier is a data point that stands out as very different from the normal range. In our example, unusual purchase quantities (far above or below the typical amount) were identified as outliers. This helps pinpoint rare or unexpected events in data.

- **Why it Matters in Retail:** In a supermarket or retail setting, finding outliers can unveil potential issues or opportunities. For instance, a sudden spike in purchases of a product could indicate a popular promotion (or a reporting error), while an abnormally low sales day might reveal a supply problem. By catching these anomalies, businesses can investigate and respond – reducing fraud, correcting errors, or capitalizing on trends[7].
 - **How MLTK Helps:** Splunk's Machine Learning Toolkit makes outlier detection beginner-friendly by providing a guided workflow (no heavy math or coding needed). The Smart Outlier Detection assistant uses your historical data as a baseline of “normal” behavior and then flags points that deviate significantly from that baseline. It assumes your past data represents normal conditions[8] – so the better your historical data, the more accurate the anomaly detection.
-

Exercise 2: Time Series Forecasting

Scenario Overview

Time Series Forecasting is about using **historical trends to predict future values**. In simpler terms, we look at how something has been changing over time and then project those patterns forward. A common definition is that forecasting “uses historical data to identify patterns, which are then used to forecast how your data might behave in the future”.

In this exercise, our focus is on **monthly sales** data. Imagine you have sales figures for each month over several years – forecasting will help you estimate what future months might look like. This is extremely relevant for business planning: if you can anticipate an **upcoming rise or drop in sales**, you can make informed decisions (like stocking inventory, staffing, budgeting, etc.). For example, if the data shows every December sales jump by 30%, a forecast can capture that seasonal pattern and predict a similar jump next December, allowing the business to prepare accordingly.

Step-by-Step Instructions

1. **Navigate to the Forecasting Example:** In the MLTK app, go back to the **Examples>Showcase** section. This time, find the category for **Forecasting**. Look for an example titled “**Forecast Monthly Sales.**” (This example uses a sample dataset of souvenir shop sales by month, which is perfect for practicing monthly sales forecasting.) Click on **Forecast Monthly Sales** to begin setting up the experiment.
2. **Create a New Forecasting Experiment:** After selecting the example, you will initiate the Time Series Forecasting assistant. Choose the **Forecast Time Series** assistant (the tool for predicting future values from time-based data) and click **Create New Experiment**. When prompted, enter a name such as “**Monthly Sales Forecast Experiment**” and then click **Create** to open the assistant interface.

3. **Load the Sales Dataset:** In the **Define** stage, select the dataset for the monthly sales. The example dataset “**souvenir_sales.csv**” will be automatically populated, representing monthly souvenir sales figures[10]. The data preview should show a date field (e.g. Month or Date) and a sales figure for each period. Ensure the time column (month) is recognized properly as a time field by the assistant (it usually is, since it’s a showcase example).
4. **Set Forecast Parameters (Optional):** For a basic run, the default settings will be used by the forecasting assistant. By default, the assistant will choose a forecasting algorithm and a forecast horizon (how far into the future to predict). For this beginner exercise, you **do not need to adjust anything** – however, note that you *can* tweak parameters. For instance, you could specify how many months ahead to predict or adjust the confidence interval. These parameters can affect the model’s results, but the default will already attempt to detect any trend or seasonality in the data. (We’ll run with defaults now, but keep in mind that in real scenarios, tuning these settings can improve accuracy.)
5. **Run the Forecast:** Click the **Forecast** (or **Run/Next**) button to train the model and generate the forecasted values. The MLTK will use your historical monthly sales data to build a forecasting model. This might involve statistical techniques (like a state-space model or ARIMA under the hood) to find patterns such as trends (overall growth/decline) and seasonality (regular ups and downs) in your data. The model will then extrapolate those patterns into the future to predict upcoming sales. The processing is automatic – within seconds, you should see results.
6. **Review Forecast Results:** Once the model finishes, the **Review** or results view will display a visualization of the forecast. Typically, you’ll see a line chart: the historical sales are plotted, and the forecasted future sales are plotted as an extension of that timeline. For example, if your data went up to December 2024, the forecast might show predicted sales for January 2025 onwards. The chart often includes a **confidence interval** (shaded region or upper/lower lines) around the forecast, indicating the uncertainty range. Pay attention to features in the plot:
 7. Do you see an **upward or downward trend** continuing? (Is the line generally going up, down, or staying flat in the future?)
 8. Are there **seasonal patterns**? (For instance, regular spikes at certain intervals that the forecast has continued forward.)
 9. The actual historical data vs. forecast: the model usually shows how it fits the tail end of your historical data as well, so you can gauge if it’s capturing the pattern accurately.
10. **Interpret the Forecast:** Now, interpret what the forecast is telling you. Suppose the forecast shows an upward trend – this implies sales are expected to increase in coming months. If there’s a repeating wave shape in the line, it indicates seasonality (perhaps high sales every 12th month, etc.). Use the visualization to

answer questions like: “Are we expecting growth or decline?” and “When are the peaks and troughs likely to occur?” This step connects the analysis back to business sense – for example, if the forecast predicts a big sales increase in summer, a retailer might plan extra stock for that period.

11. **(Optional) Experiment with Tuning:** If time permits and you’re curious, you can try adjusting one of the parameters and re-run the forecast to see the effect. For instance, you might change the forecast horizon to predict farther out, or if the assistant allows, adjust the seasonal period detection. (This is optional in this lab, but it’s good to know that **forecasting models often require tuning** – different settings or algorithms can yield different results, and finding the best fit may require a few tries.)
12. **Conclude the Forecasting Exercise:** You have now performed a time series forecast of monthly sales using Splunk MLTK. Take note of the key findings from the forecast. In a real-world scenario, this forecast could be used to make decisions (for example, if the forecast indicates a dip in sales next month, a company might launch a marketing campaign to boost demand, or if a big rise is expected, they might increase inventory). Discuss how such forecasting ability can benefit planning and strategy.

Summary – Key Takeaways from Time Series Forecasting

- **What is Time Series Forecasting?** It is a method to predict future data points by analyzing past time-ordered data. Essentially, the model learns patterns from historical trends and projects those patterns forward in time. In our case, we used past monthly sales to forecast future monthly sales.
- **Why it's Useful:** Forecasting gives businesses a **heads-up** about what's likely to happen. By knowing the possible future trend (e.g., expecting higher sales in holiday season or a slow period in winter), organizations can plan better. For example, if sales are forecasted to rise 20% next quarter, a company can stock more products and schedule more staff. If a downturn is forecasted, they might tighten budgets or ramp up marketing. Forecasts turn historical insights into forward-looking **strategic planning tools**.
- **Assumptions and Data:** Keep in mind that forecasts assume the future will behave somewhat like the past. The model we used assumes the historical data is a good baseline and that existing patterns (trend or seasonality) will continue. That's why having reliable, sufficient historical data is important – the better the quality of past data, the more confidence we have in the forecast. Unforeseen events (like a sudden market change) can make actual future values differ from the forecast, so always use forecasting as a guide, not a guaranteed outcome.
- **Tuning and Parameters:** The Machine Learning Toolkit provides parameters you can adjust to refine your forecasts. For instance, you can change how many future points to predict, set the period of seasonality, or choose different algorithms (if available). These parameters have a **big impact on the results** – tuning them can

improve accuracy. In practice, analysts may try several models or parameter settings and compare results to pick the best forecast. For our beginner exercise, we used defaults, but it's good to know that MLTK allows this flexibility as you become more comfortable with the tool.

By completing these two exercises, you've gotten a hands-on introduction to Splunk's Machine Learning Toolkit. You detected outliers in historical data and generated a forecast for future data – two powerful techniques in data analysis. With these basics, you can start exploring more complex scenarios, always remembering that machine learning in Splunk is about leveraging your data (past and present) to gain actionable insights for the future.

Happy Splunking! [1]

[1] [7] What Is Anomaly Detection? Examples, Techniques & Solutions | Splunk

https://www.splunk.com/en_us/blog/learn/anomaly-detection.html

[2] [3] [5] [6] Smart Outlier Detection Assistant | Splunk Docs

<https://help.splunk.com/en/splunk-enterprise/apply-machine-learning/use-ai-toolkit/5.6.0/smart-assistant-guided-workflows/smart-outlier-detection-assistant>

[4] [10] Splunk Machine Learning Toolkit Showcase | Splunk Docs

<https://help.splunk.com/en/splunk-cloud-platform/apply-machine-learning/use-ai-toolkit/5.6.3/introduction-to-the-splunk-machine-learning-toolkit/splunk-machine-learning-toolkit-showcase>