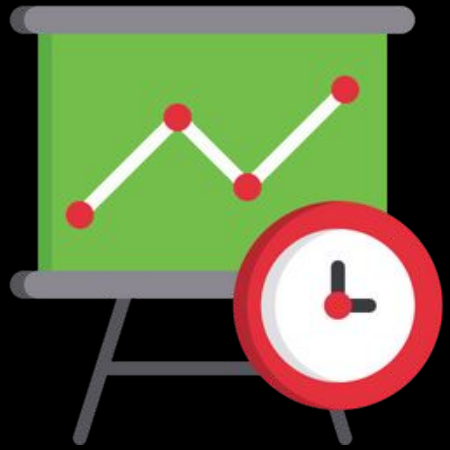
Time Series Analysis



Aakanksha Duggal
Hema Veeradhi

Emerging Tech, Office of the CTO

Hema Veeradhi
Senior Data Scientist

Sunnyvale, CA, USA

Aakanksha Duggal
Data Scientist

Boston, MA, USA

Emerging Tech, Office of the CTO

Index

- Introduction (Hema&Aakanksha) - 1-3min
- What is time series? (Aakanksha) - 5min
- Real world Applications (Aakanksha) - 5min
  - Customer growth prediction (sales)
  - Anomaly detection
- Algorithms (1-5min each)
  - Linear Regression (Aakanksha)
  - Exponential (Aakanksha)
  - ARIMA (Hema)
  - SARIMA (Hema)
  - Prophet (in PAD) (Hema)
- Example - PAD (Hema) (5min)
- Resources - time series repo, pad , cloud price analysis (Hema)

What is Time Series?

A ***time series*** is nothing but a sequence of various data points that occurred in a successive order for a given period of time.

Objectives

- **Factors**
- **Consequences and insights of features**
- **Predicting the future**

Components

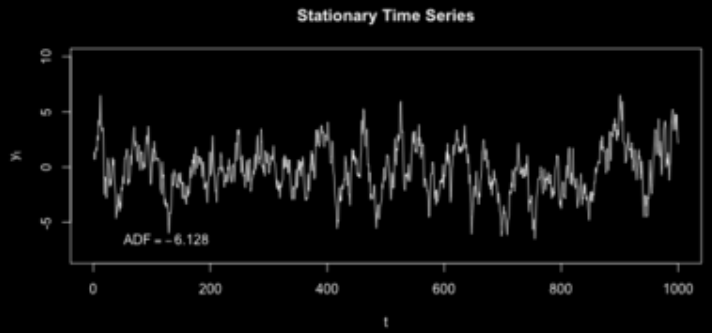
- **Trend**
- **Seasonality**
- **Cyclical**
- **Irregularity**

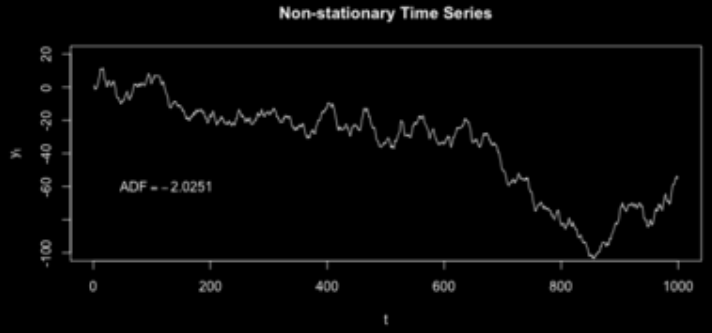
Significance

- **Forecasting**
- **Segmentation**
- **Classification**
- **Descriptive analysis`**
- **Intervention analysis**

What is Time Series?

A ***time series*** is nothing but a sequence of various data points that occurred in a successive order for a given period of time.



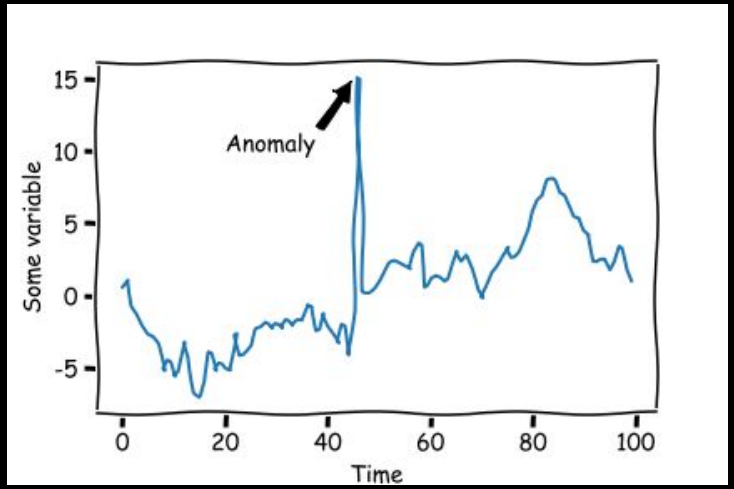


Types of Time Series Data

- **Stationary**
- **Non-Stationary**

Applications

Customer Sales Forecasting



Anomaly Detection

Source :
https://medium.com/@Momentum_London/sales-forecasting-8c5a0b61ac5a
https://dhi.github.io/tsod/

Some Basic Forecasting Algorithms

Average Method

The forecasts of all future values are equal to the average (or “mean”) of the historical data.

Naive Method

All forecasts are set to be the value of the last observation.

Seasonal Naive Method

In this case, we set each forecast to be equal to the last observed value from the same season (e.g., the same month of the previous year).

Drift Method

A variation on the naïve method is to allow the forecasts to increase or decrease over time, where the amount of change over time (called the drift) is set to be the average change seen in the historical data.

Linear Regression

The regression model allows for a linear relationship between the forecast variable y and a single predictor variable x.

$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t.$$

Exponential Smoothing

Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older.

Simple Exponential Smoothing
The simplest of the exponentially smoothing methods is naturally called simple exponential smoothing. This method is suitable for forecasting data with no clear trend or seasonal pattern.

Trend(Holt) Methods
Holt extended simple exponential smoothing to allow the forecasting of data with a trend. This method involves a forecast equation and two smoothing equations (one for the level and one for the trend)

Holt-Winter's Seasonal Models
Extension of the Holt model to incorporate seasonality. The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations — one for the level ℓ_t, one for the trend b_t, and one for the seasonal component s_t, with corresponding smoothing parameters α, β^* and γ. We use m to denote the frequency of the seasonality, i.e., the number of seasons in a year.

ARIMA

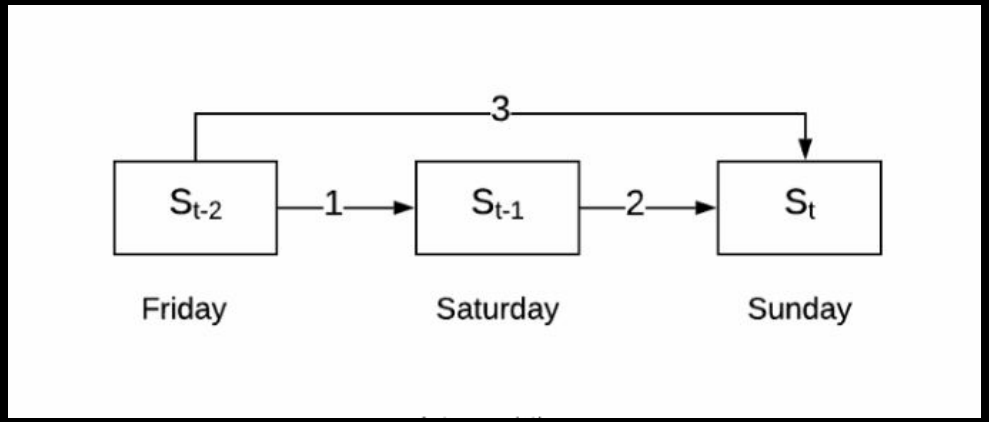
- ARIMA, short for '**Auto Regressive Integrated Moving Average**' is actually a class of models that 'explains' a given time series based on its own past values
- ARIMA model assumes that the time series stationary i.e. is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time

Components of ARIMA:

- ❖ AR: Autoregression - A model that uses the dependent relationship between an observation and some number of lagged observations.
- ❖ I: Integrated - The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
- ❖ MA: Moving Average - A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

ARIMA

S_t = Price of Gas on Sunday
S_{t-1} = Price of Gas on Saturday
S_{t-2} = Price of Gas on Friday



Auto-correlation

This mutual relationship between day prices here, is called correlation. If values of two variables **increase (or decrease) together** then they are said to have a **positive correlation**. If the value of **one variable increases and the other decreases (or vice versa)** then they have a **negative correlation.**

ARIMA

Correlation(Sunday,Saturday) => lag factor of 1
Correlation(Sunday, Friday) => lag factor of 2
Correlation(Sunday, Thursday) => lag factor of 3

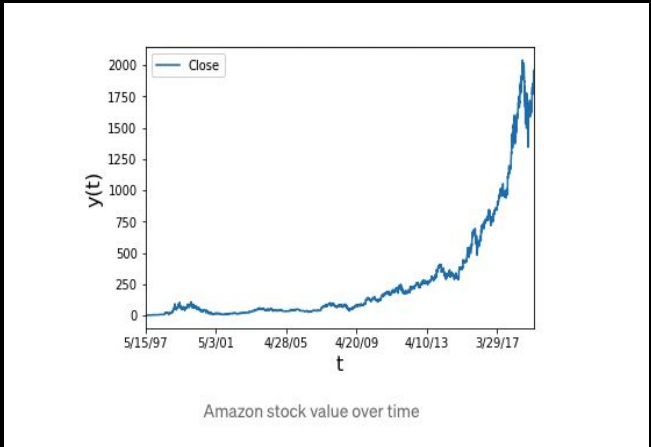
Auto Regressive (AR) model is a specific type of regression model where, the dependent variable depends on past values of itself.

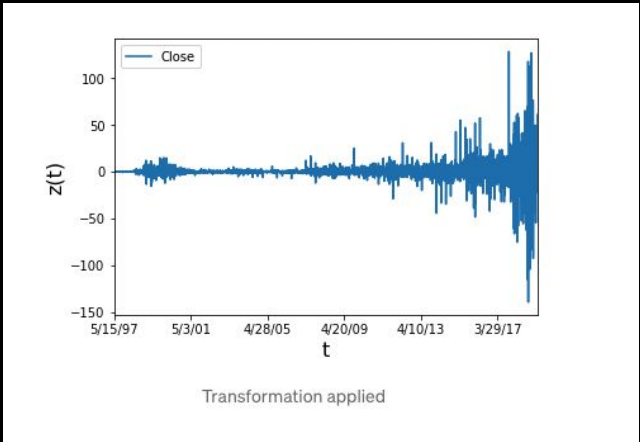
$$Y_t = \beta_1 + \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \ldots + \Phi_p Y_{t-p}$$

AR Equation

ARIMA

- Lets say, for example, you come across a series with a “non-constant” mean. Its clearly observable that the mean increases over time i.e. the series is not stationary.





- **I stands for Integrated** (though it has nothing to do with integration). It just means that, instead of predicting the time series itself we will predict the differences of the series from one time step to the next time step.

ARIMA

- **Moving Average (MA)** model works by analysing how wrong you were in predicting values for the previous time-periods to make a better estimate for the current time-period.
- MA model supplements the AR model by taking into considerations, the errors from the previous time-periods thereby helping to get a better estimate.

So to revise, the final ARIMA model will take the following form:

ARIMA(p, d, q) where,

p → lag order
d → order of differencing
q → size of moving average window

SARIMA

- An **extension to ARIMA** that supports the direct modeling of the **seasonal component of the series** is called SARIMA.
- It adds three new hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

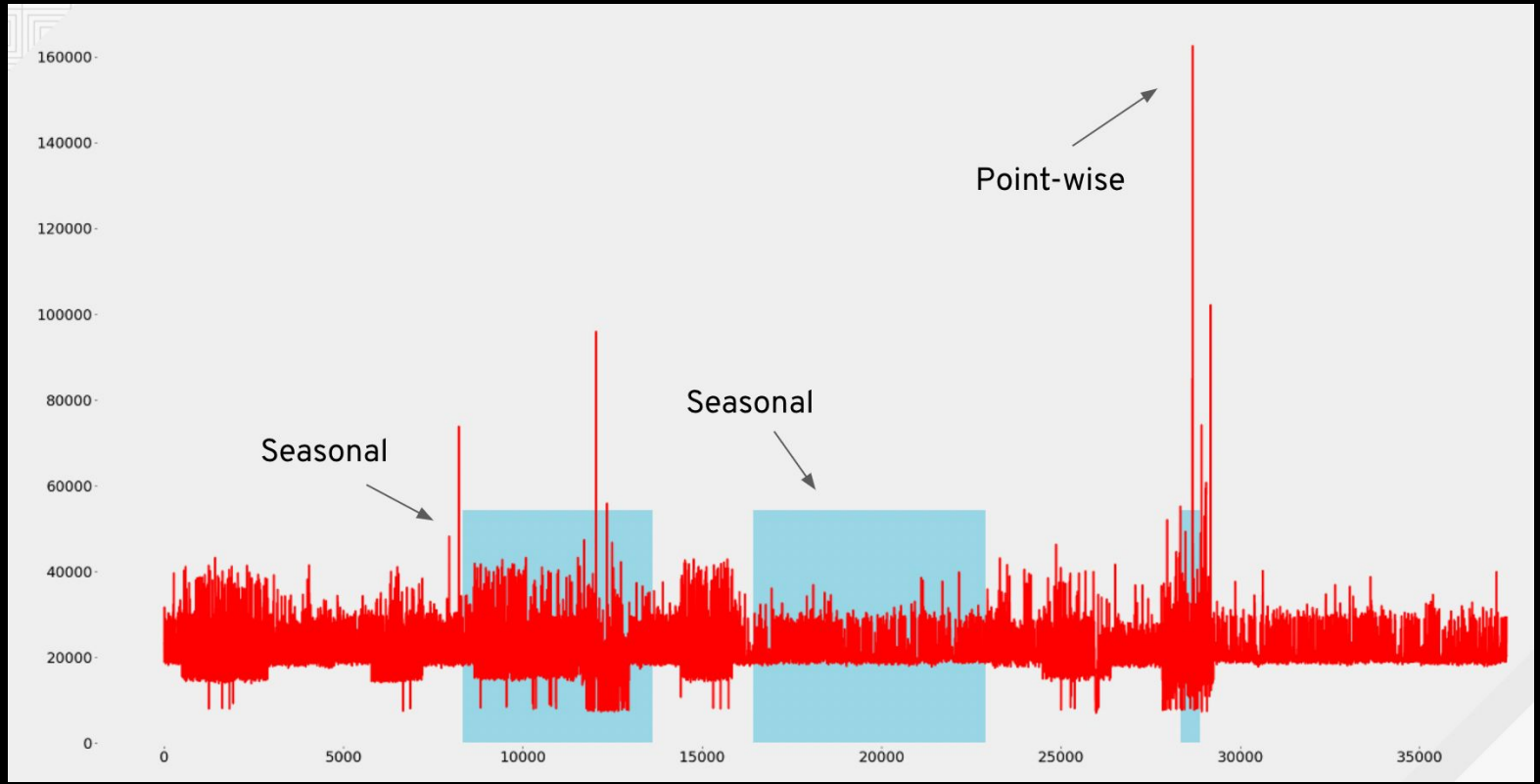
Seasonal Elements:

There are four seasonal elements that are not part of ARIMA that must be configured; they are:

P: Seasonal autoregressive order
D: Seasonal difference order
Q: Seasonal moving average order
m: The number of time steps for a single seasonal period

Together, the notation for an SARIMA model is specified as:

SARIMA(p, d, q)(P, D, Q)m


160000
140000
120000
100000
80000
60000
40000
20000
0
0
5000
10000
15000
20000
25000
30000
35000
Point-wise
Seasonal
Seasonal

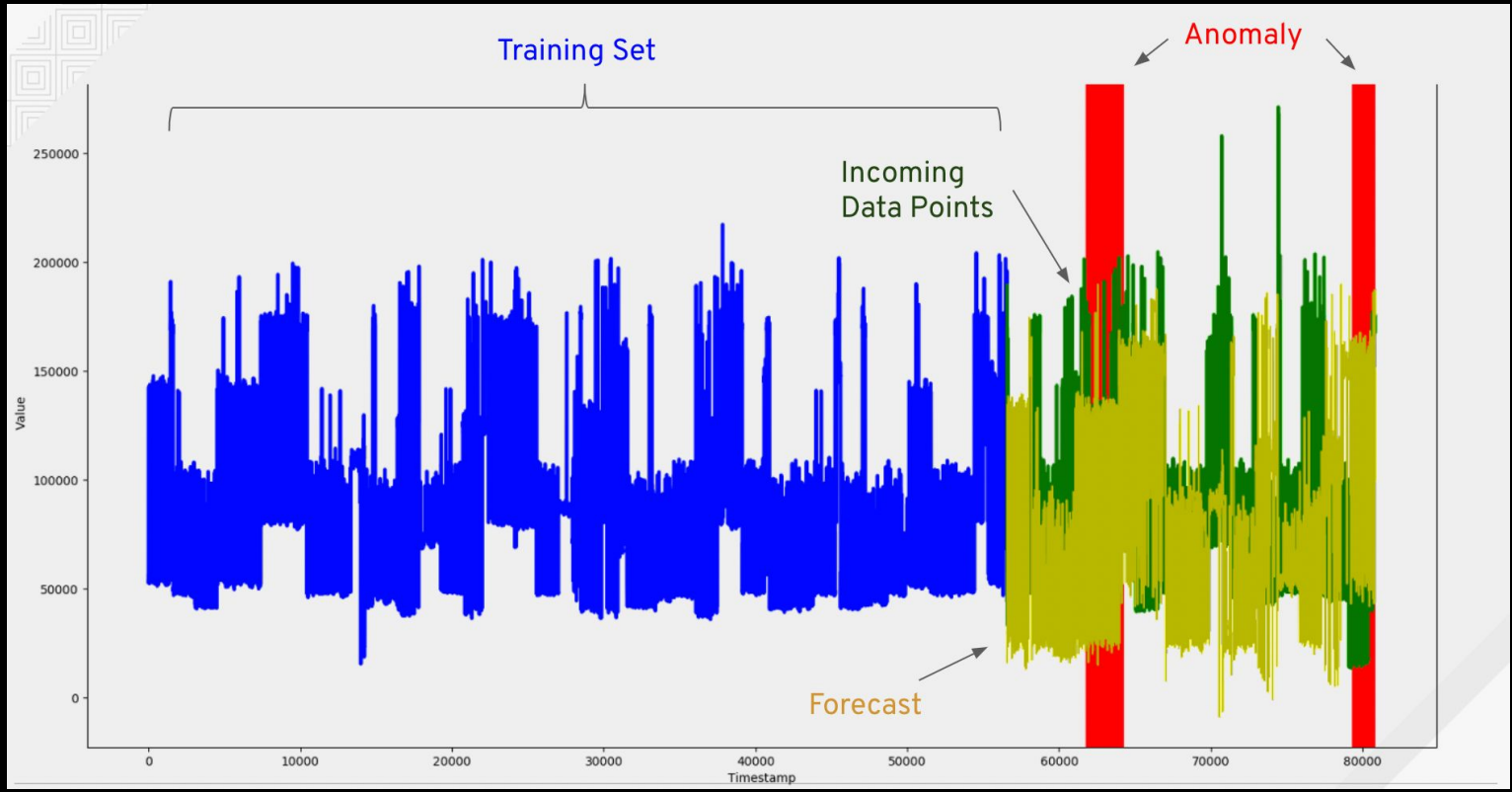
Prophet

- Prophet is open source software released by Facebook's Core Data Science team. It is a procedure for forecasting time series data based on an **additive model** where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data.

- At its core is the sum of three functions of time plus an error term: **growth g(t)**, **seasonality s(t)**, **holidays h(t)** , and **error e_t** :

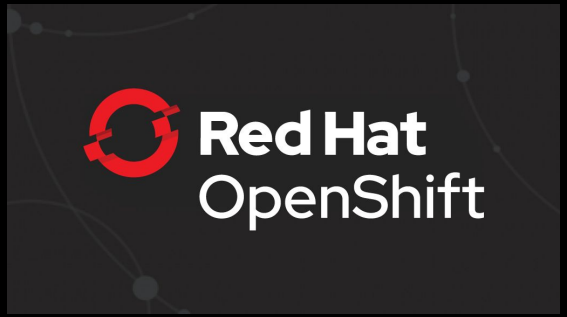
$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

Prophet


Training Set
Anomaly
Incoming
Data Points
Forecast
Value
Timestamp
250000
200000
150000
100000
50000
0
0
10000
20000
30000
40000
50000
60000
70000
80000

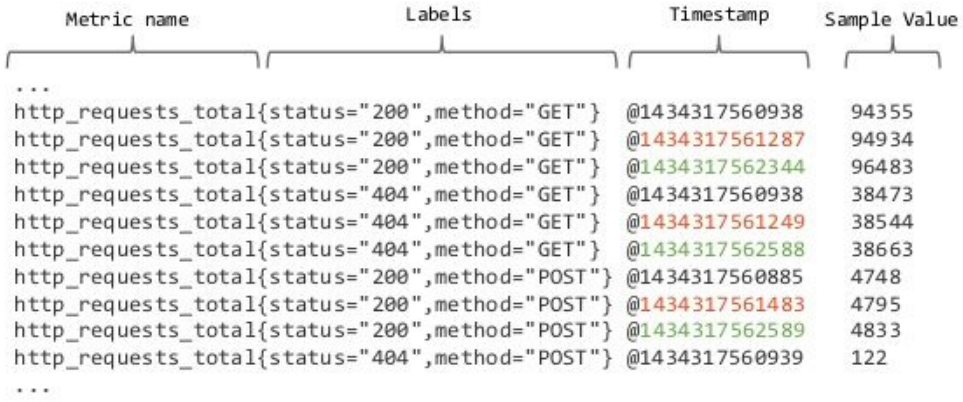
Time Series Anomaly Detection

AI-backed anomaly detection



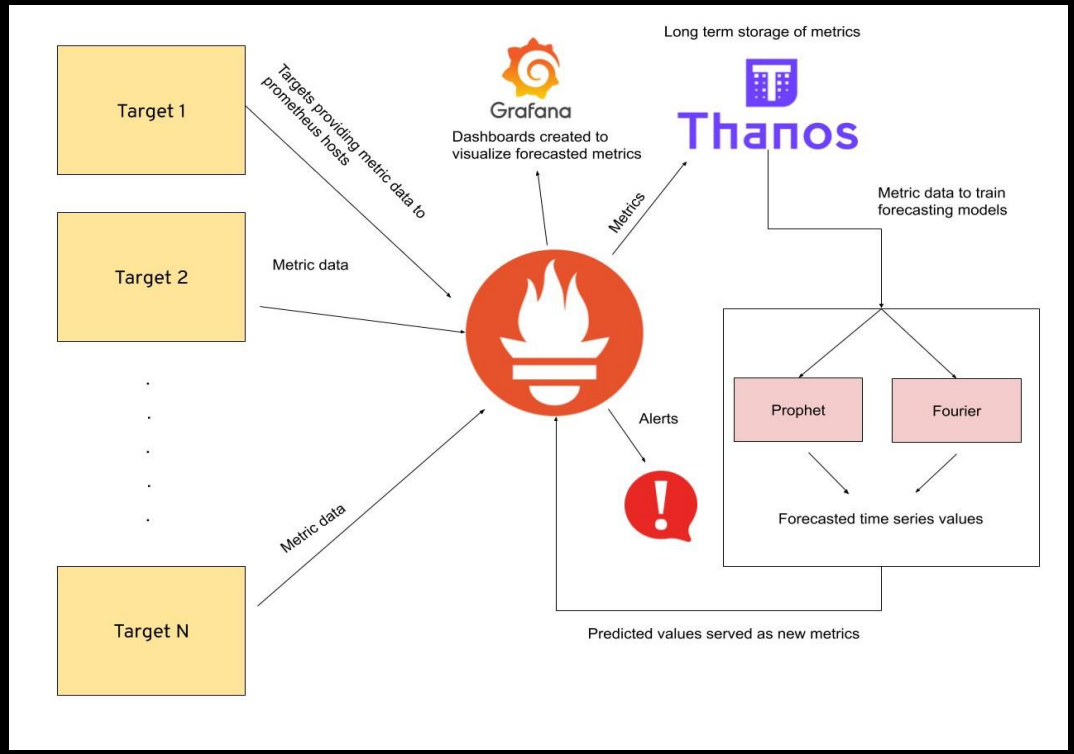
Monitoring → Applications → Prometheus

Prometheus Metrics

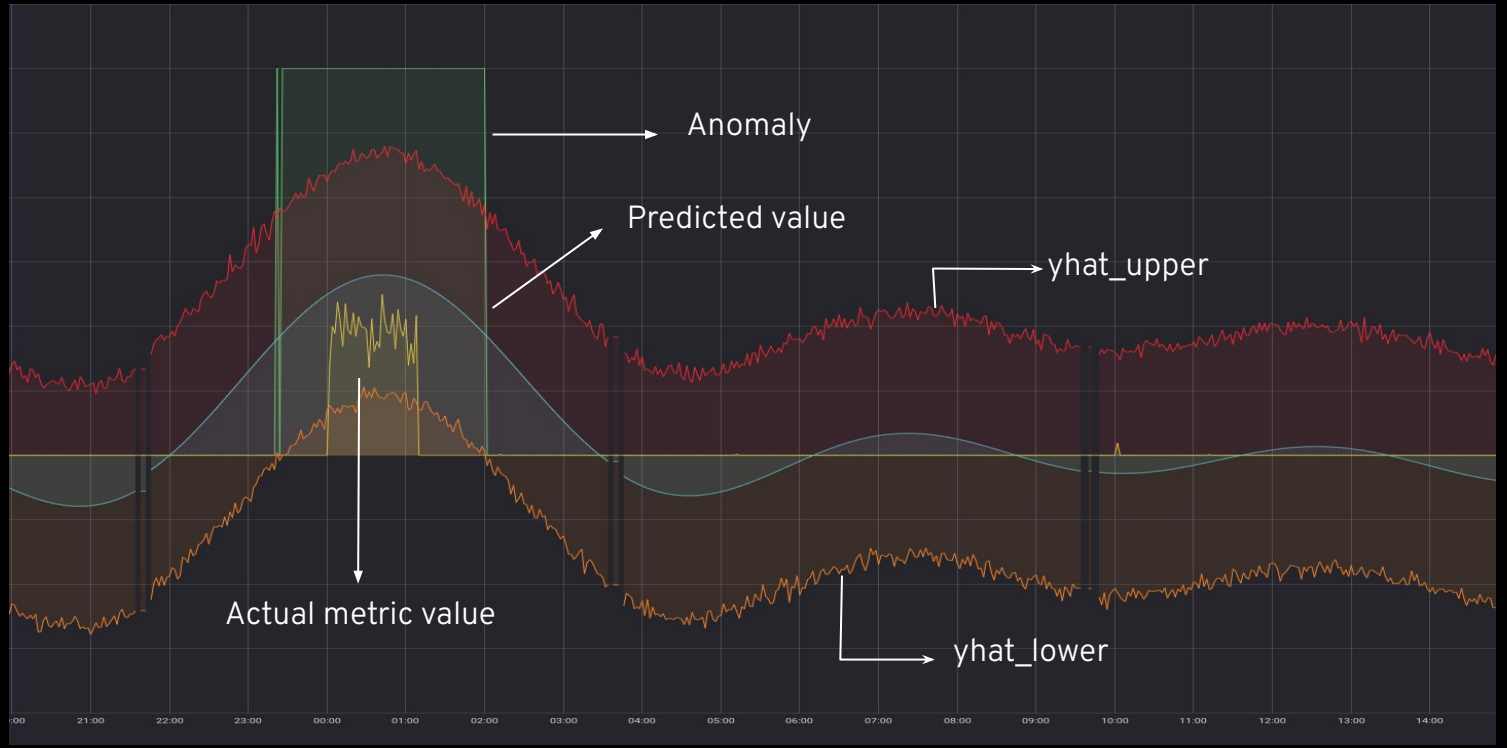


- **Gauge** - Arbitrary up and down value
- **Counter** - Monotonically Increasing
- **Histogram** - Cumulative samples of values
- **Summary** - Snapshot of values in a time window

Implementation



Grafana Dashboard



Resources

- Time Series Analysis GitHub repo - https://github.com/aicoe-aiops/time-series
- Prometheus Anomaly Detection workshop Part 1 - https://developers.redhat.com/courses/ai-ml/prometheus
- Prometheus Anomaly Detection workshop Part 2 - https://developers.redhat.com/courses/ai-ml/time-series-forecasting-prometheus-metrics
- Prometheus Anomaly Detection project - https://github.com/AICoE/prometheus-anomaly-detector
- Time Series Algorithm Notebooks - https://github.com/aicoe-aiops/cloud-price-analysis/tree/master/notebooks/experimental/Time%20Series

Thank You!