AWS Academy Machine Learning Foundations

Module 4: Introducing Forecasting

aws academy

Module overview

Sections

1. Forecasting overview
2. Processing time series data
3. Using Amazon Forecast
4. Guided lab
5. Module wrap-up

Demonstrations

- Creating a forecast with Amazon Forecast

Guided Lab

- Creating a Forecast with Amazon Forecast

Knowledge check

Module objectives

At the end of this module, you should be able to:

- Describe the business problems solved by using Amazon Forecast
- Describe the challenges of working with time series data
- List the steps that are required to create a forecast by using Amazon Forecast
- Use Amazon Forecast to make a prediction

Module 4: Introducing Forecasting

Section 1: Forecasting overview

Overview of forecasting

- Predicting future values that are based on historical data
  - Can be either univariate or multivariate
- Common patterns –
  - Trends: Patterns that increase, decrease, or are stagnant
  - Seasonal: Pattern that is based on seasons
  - Cyclical: Other repeating patterns
  - Irregular: Patterns that might appear to be random

Trending data

Seasonal data

Cyclical data

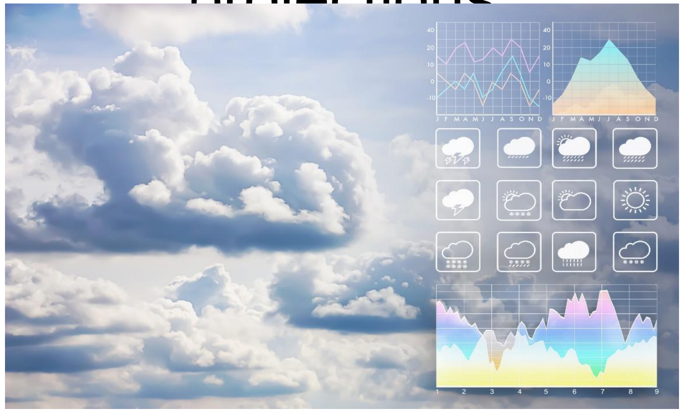
Irregular data

Forecasting use cases

Sales and demand forecasts

Inventory projections

Energy consumption



Weather forecasts

Module 4: Introducing Forecasting

Section 2: Processing time series data

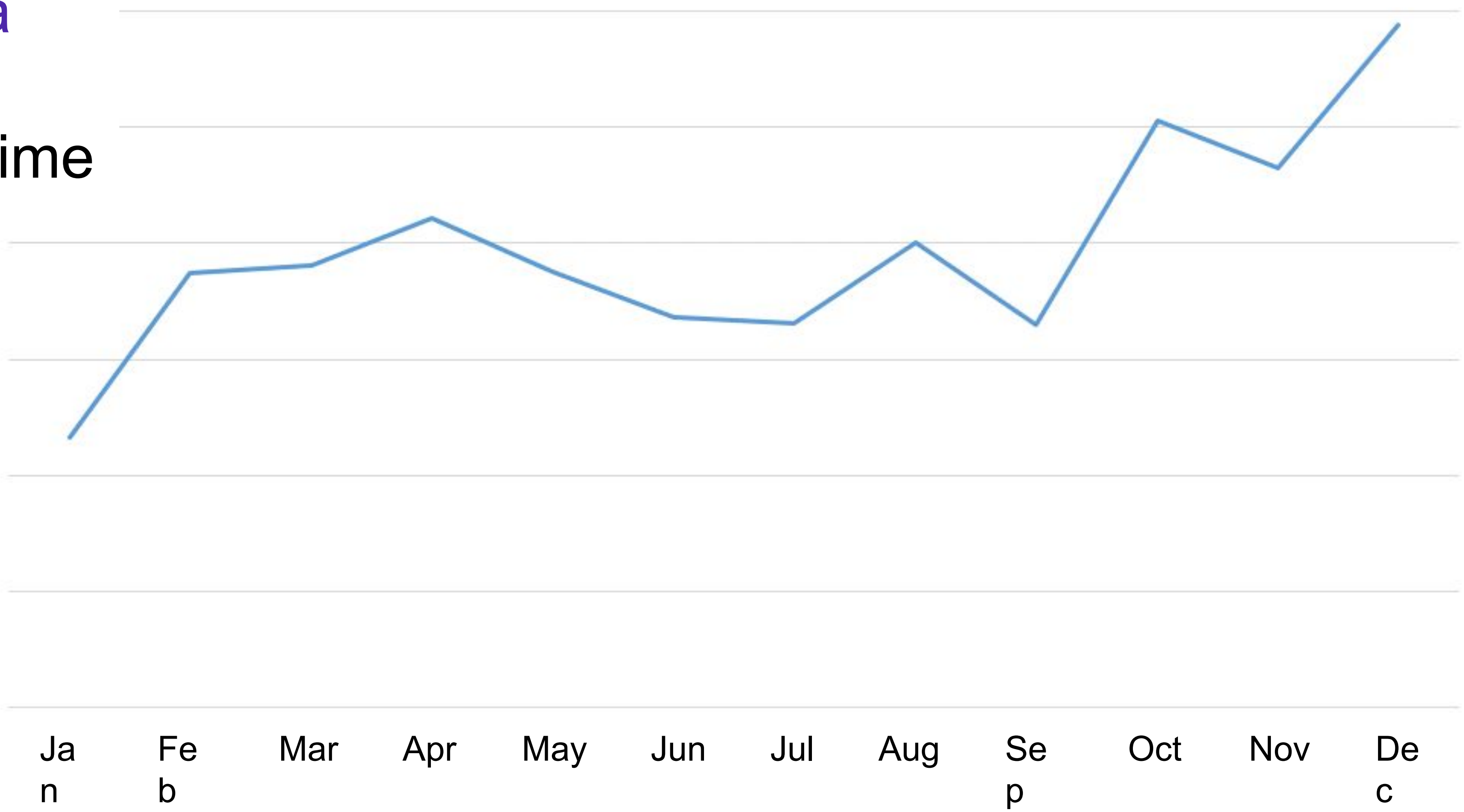
Time series data

Time series data is captured in sequence over time

Related data informs the time series data—for example, price or promotions

Metadata might also be needed to explain predictions—for example, brand name or category

Unit Sales Product #21232



Time and date challenges

Incomplete and varying timestamps

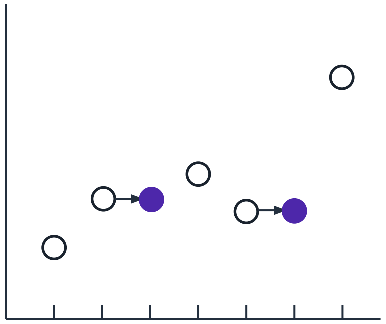
yyyy-mm-dd HH:MM:SS	Includes time
yyyy-dd-mm	Year, day, month
yyyy-mm-dd	Year, month, day
yyyy-mm	No day
ss:s	Seconds
mm-dd	No year

UTC, local, and time zones:

Is the time in UTC format?

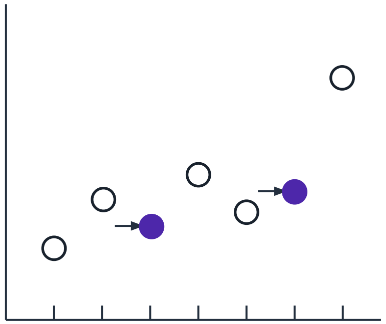
2020-06-02T13:15:30Z

Time series handling: Missing data



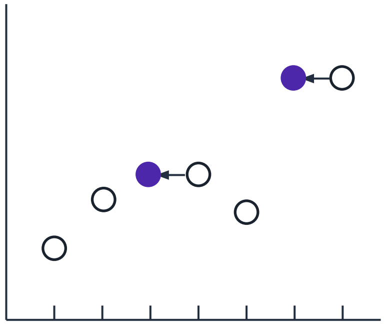
Forward Fill

Last known == missing value



Moving Average

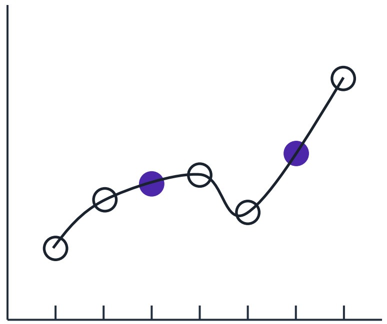
Avg(previous) == missing value



Backward Fill

Next value == missing value

Danger! Lookahead



Interpolation

Linear
Spline
Polynomial

Note: Zero is sometimes the perfect fill value

Time series handling: Downsampling

InvoiceDate (yyyy-mm-dd HH:MM:SS)	Item	Quantity
2009-12-01 07:45:00	21232	24
2009-12-01 10:06:00	21232	36
2010-12-08 14:21:00	21232	17
2010-12-08 13:11:00	21232	9
2010-12-09 18:28:00	21232	44

→ Mean/ Sum →

InvoiceDate (yyyy-mm-dd)	Item	Quantity
2009-12-01	21232	60
2010-12-08	21232	26
2010-12-09	21232	44

Time series handling: Upsampling

InvoiceDate (yyyy-mm-dd)	Item	Quantity
2009-12-01	21232	60
2010-12-08	21232	26
2010-12-09	21232	44

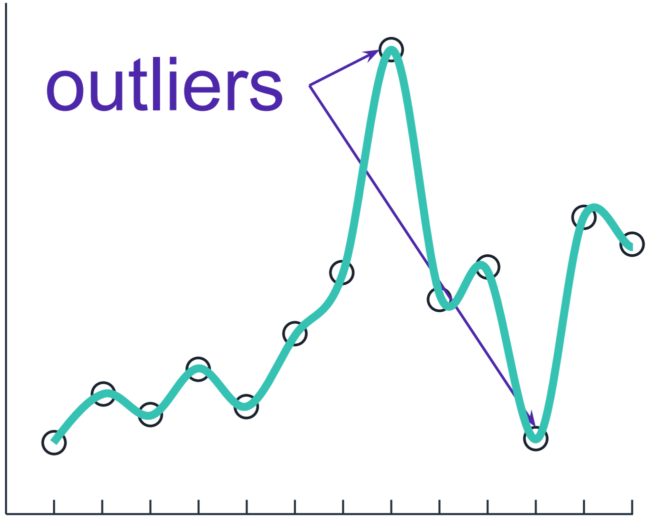
→ ? →

InvoiceDate (yyyy-mm-dd HH:MM:SS)	Item	Quantity
2009-12-01 ??:??:??	21232	??
2009-12-01 ??:??:??	21232	??
2010-12-08 ??:??:??	21232	??
2010-12-08 ??:??:??	21232	??
2010-12-09 ??:??:??	21232	44

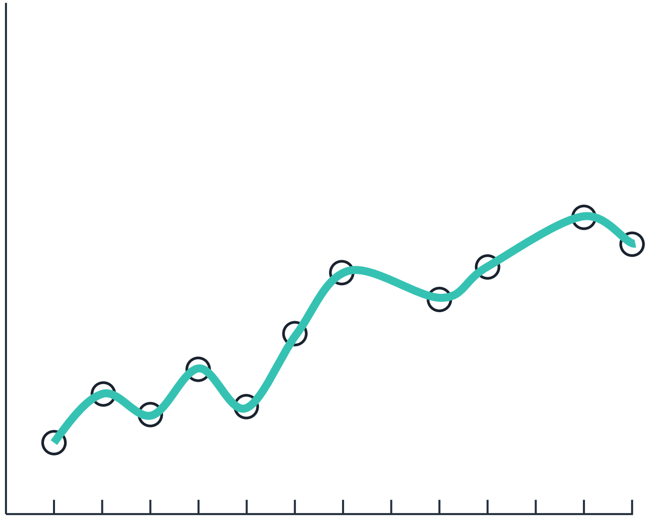
Reasons to upsample:

- Match different time series
- Irregular time series
- Knowledge of domain

Time series handling: Smoothing data



Smoothing function



Why are you smoothing?

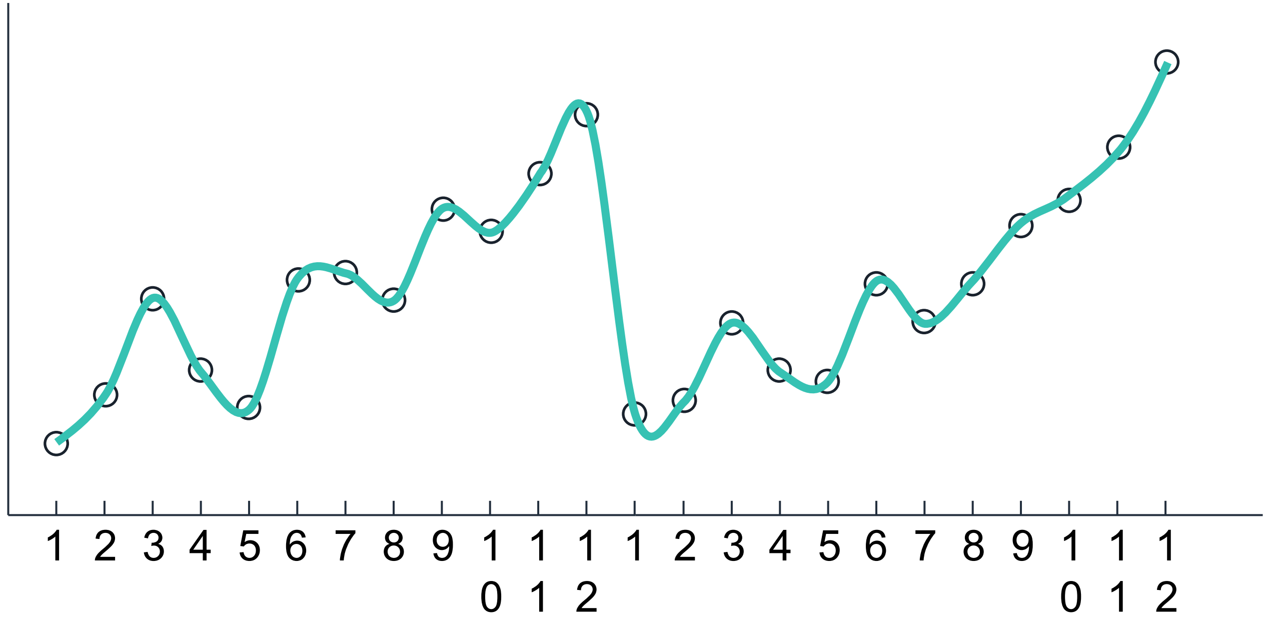
- Data preparation
- Visualization

How does smoothing affect your outcome?

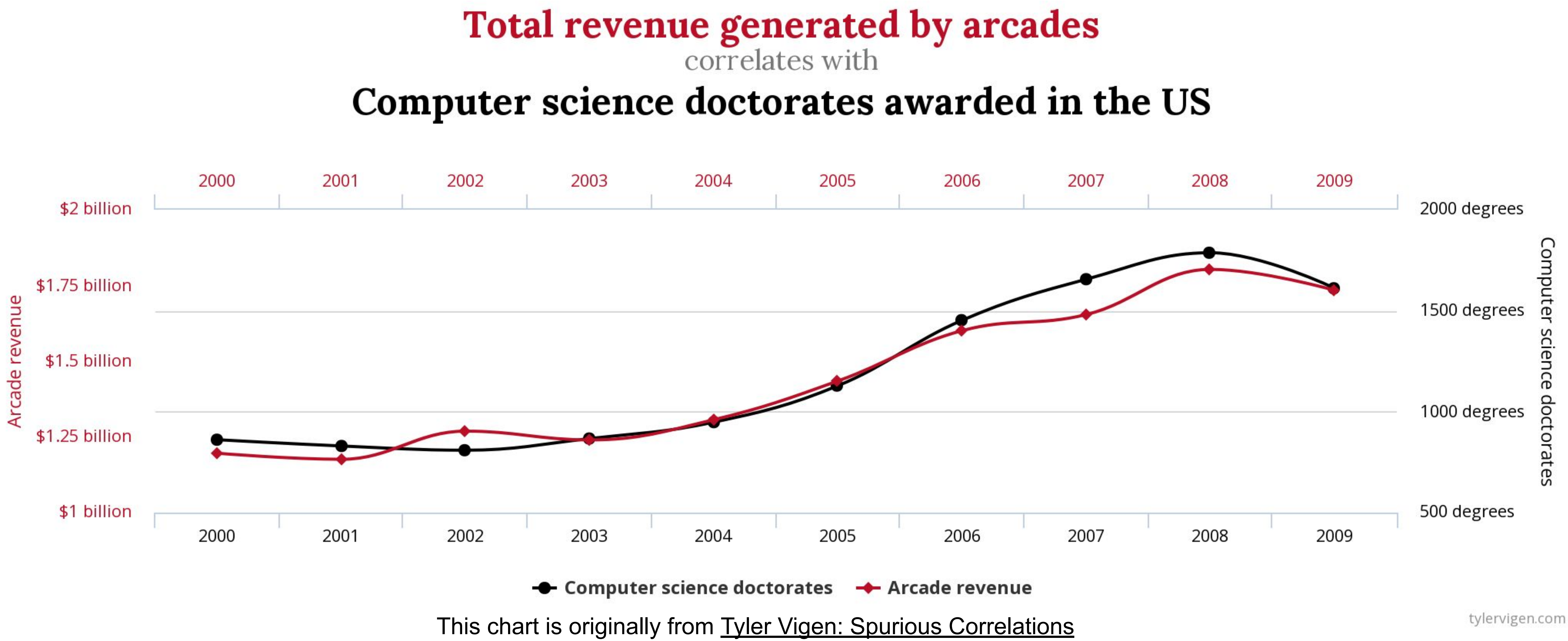
- Cleaner data to model
- Model compatibility
- Production improvements

Seasonality

- Seasonality frequency
  - Hourly, daily, quarterly, yearly
  - Spring, summer, fall, winter
  - Major holiday sales, winter holiday season
- Incorporating holidays



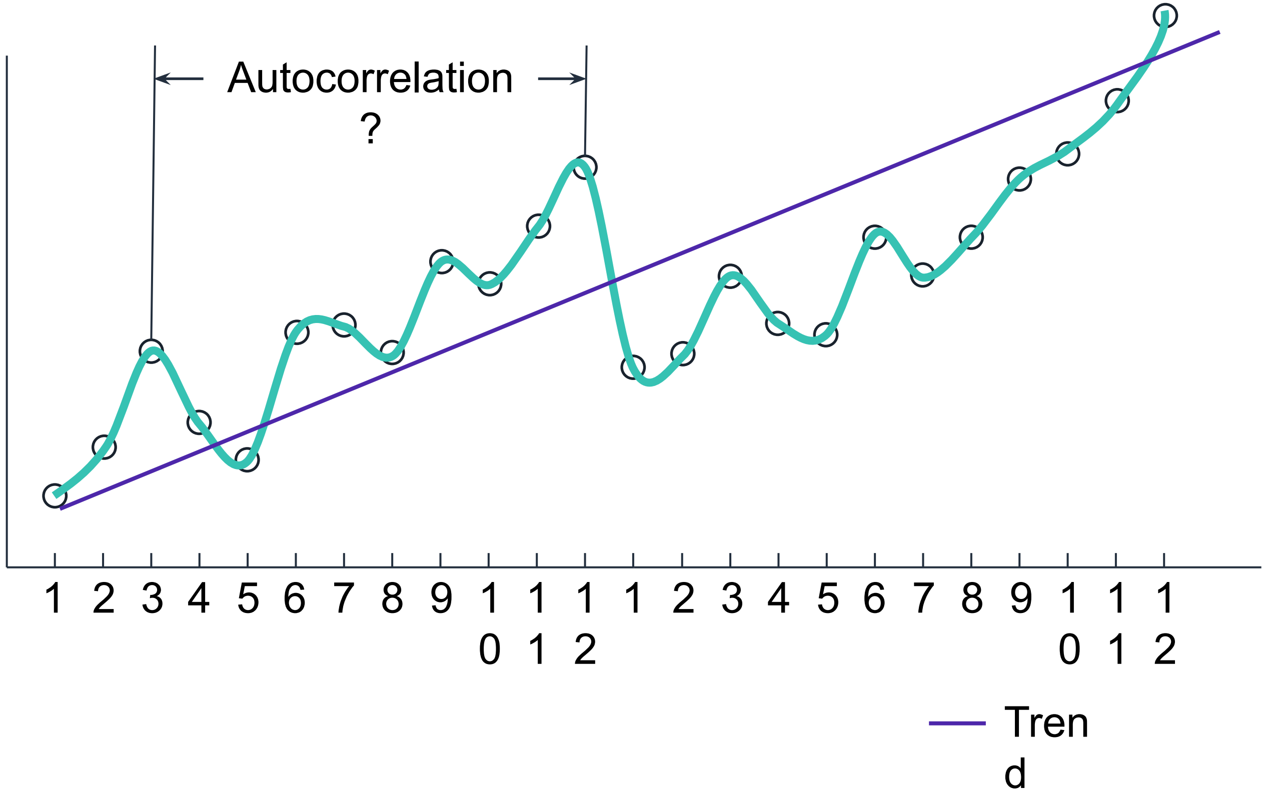
Time series correlations



This chart is originally from [Tyler Vigen: Spurious Correlations]

Stationarity, trends, and autocorrelation

- Stationarity
  - How stable is the system?
  - Does the past inform the future?
- Trends
  - Correlation issues
- Autocorrelation
  - How points in time are linearly related
- Influences algorithm choice



Using pandas for time series data

- Time-aware index

```
dataframe['2010-01-04']
```

```
dataframe['2010-02':'2010-03']
```

```
dataframe['weekday_name'] = dataframe.index.weekday_name
```

- GroupBy and resampling operations

```
dataframe.groupby('StockCode')
```

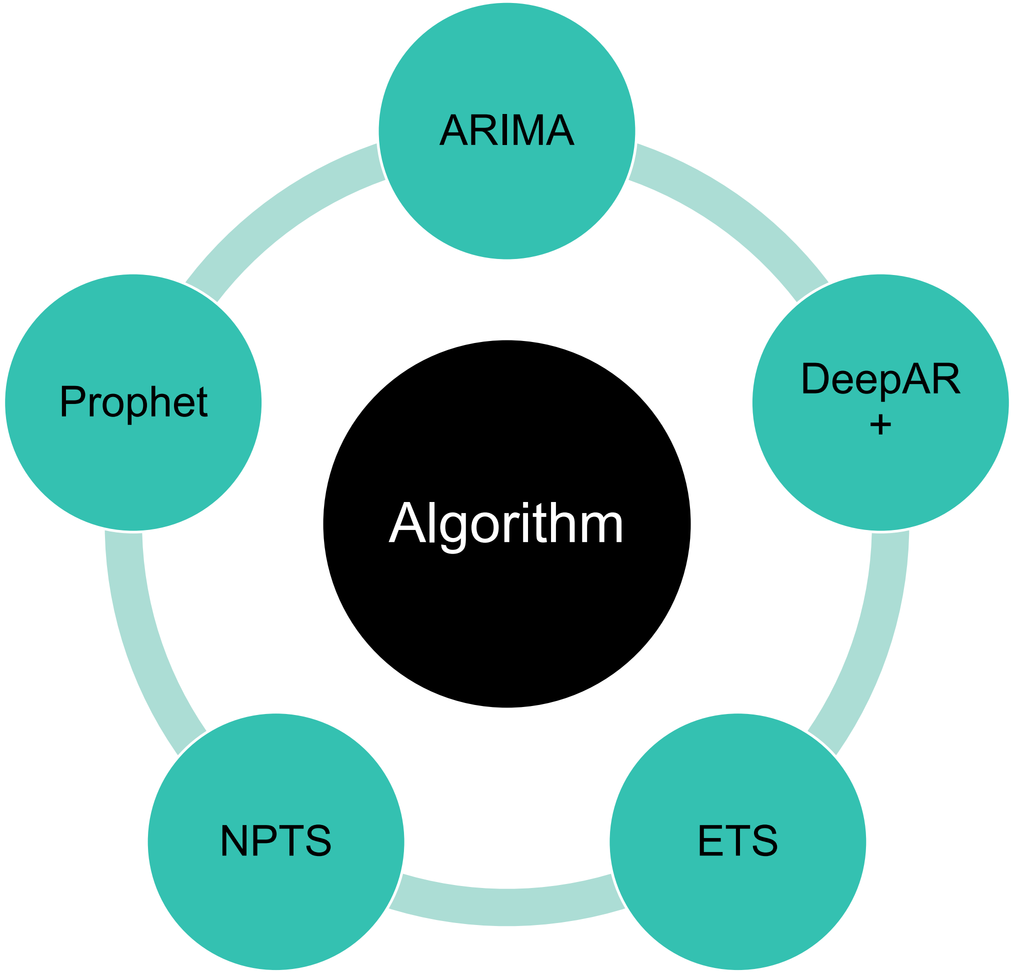
```
dataframe.groupby('StockCode').resample('D').sum()
```

- Autocorrelation

```
dataframe['Quantity'].autocorr()
```

Time series algorithms

- Autoregressive Integrated Moving Average (ARIMA)
- DeepAR+
- Exponential Smoothing (ETS)
- Non-Parametric Time Series (NPTS)
- Prophet



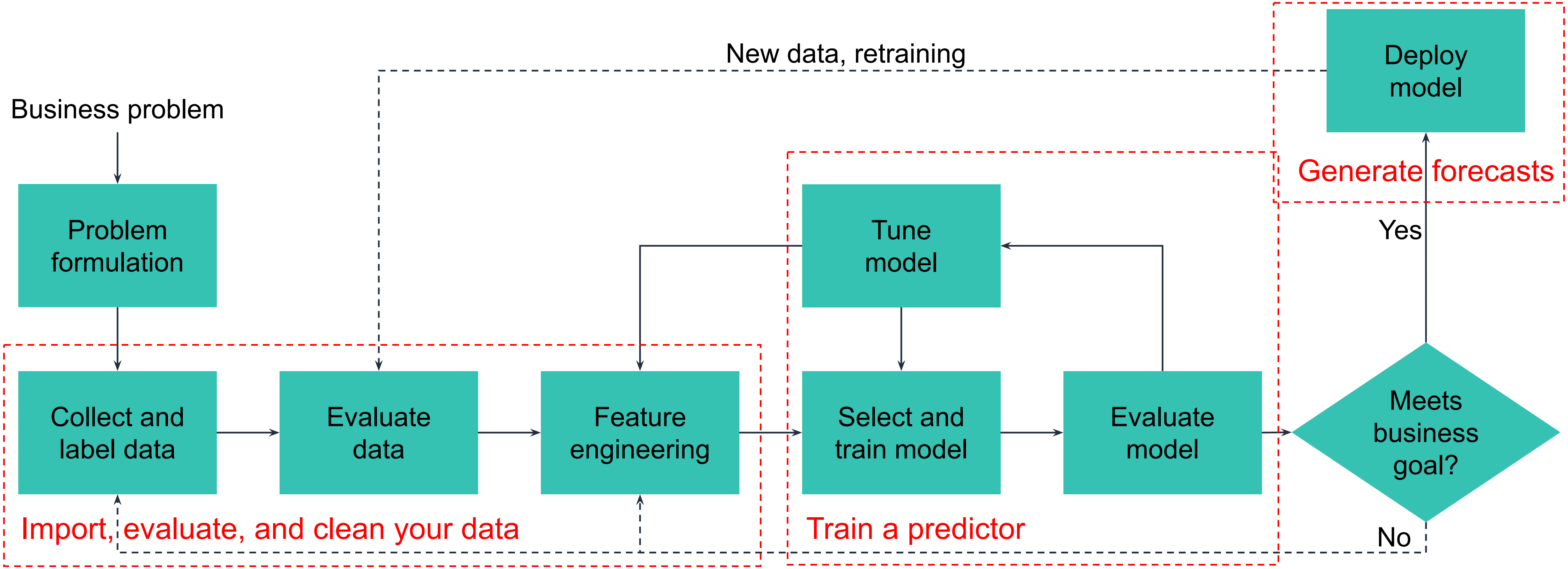
Section 2 key takeaways

- Time series data is sequenced
- Time challenges –
  - Different formats
  - Missing data
  - Seasonality
  - Correlations
- The pandas library offers support for time series data
- With Amazon Forecast, you can choose between five algorithms

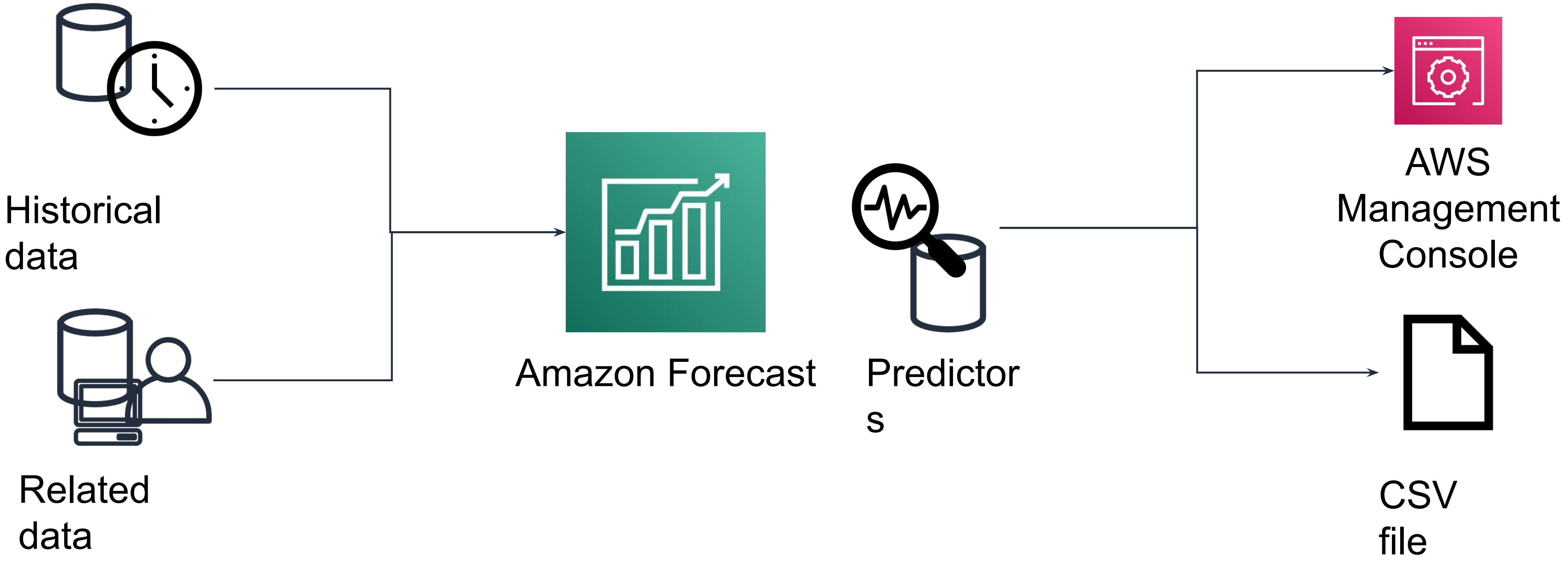
Module 4: Introducing Forecasting

Section 3: Using Amazon Forecast

Amazon Forecast workflow

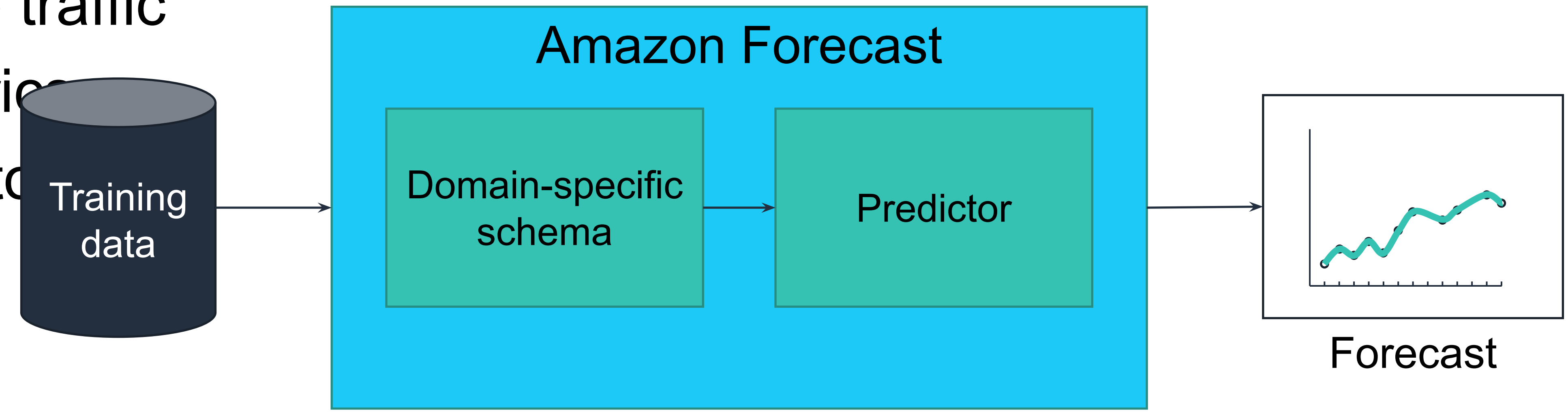


Amazon Forecast overview



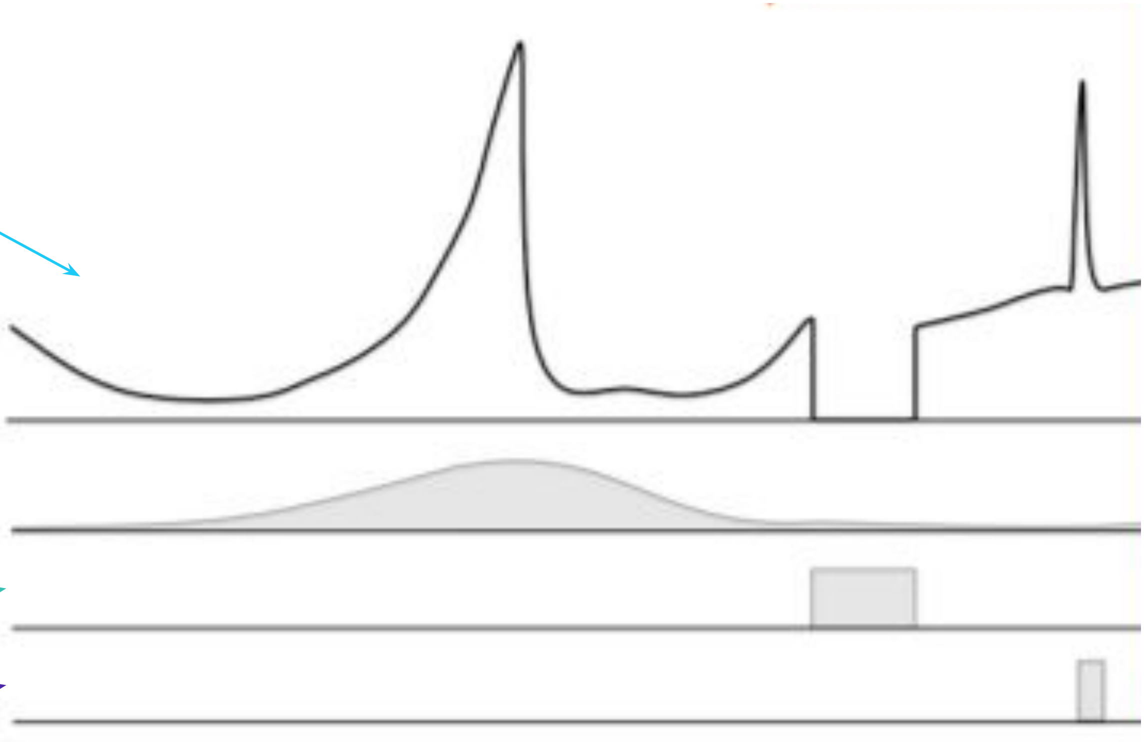
Supported domains

- Retail
- Inventory planning
- Amazon EC2 capacity
- Work force
- Web traffic
- Metric
- Custo



Retail forecasting example

- Time series data
  - Transactional sales data
  - Timestamp, item, quantity
- Metadata
  - Category, item color
  - Item, metadata
- Related data
  - Time series
  - In-stock data
  - Promotion data
  - Timestamp, item, price



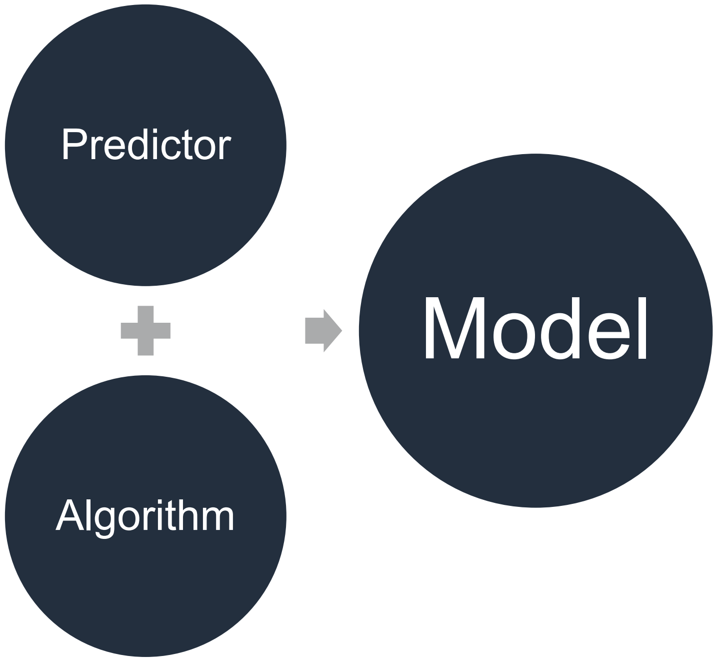
Web traffic forecast example

- Time series data
  - Webpage ID
  - Page views per month
  - Timestamp
- Related and metadata
  - Page category
  - Geographic identifier

Selecting an Amazon Forecast algorithm

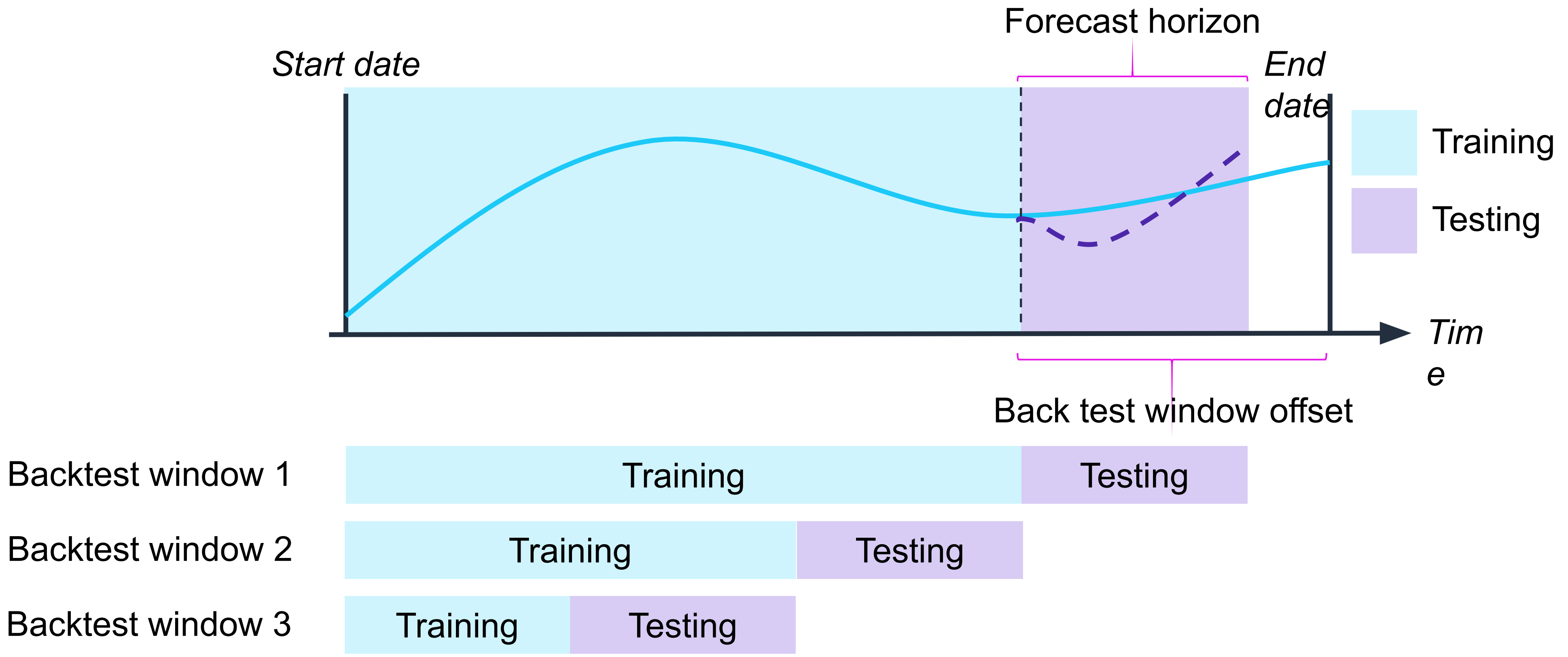
You can select from the following list of algorithms:

- ARIMA
- DeepAR+
- ETS
- NPTS
- Prophet



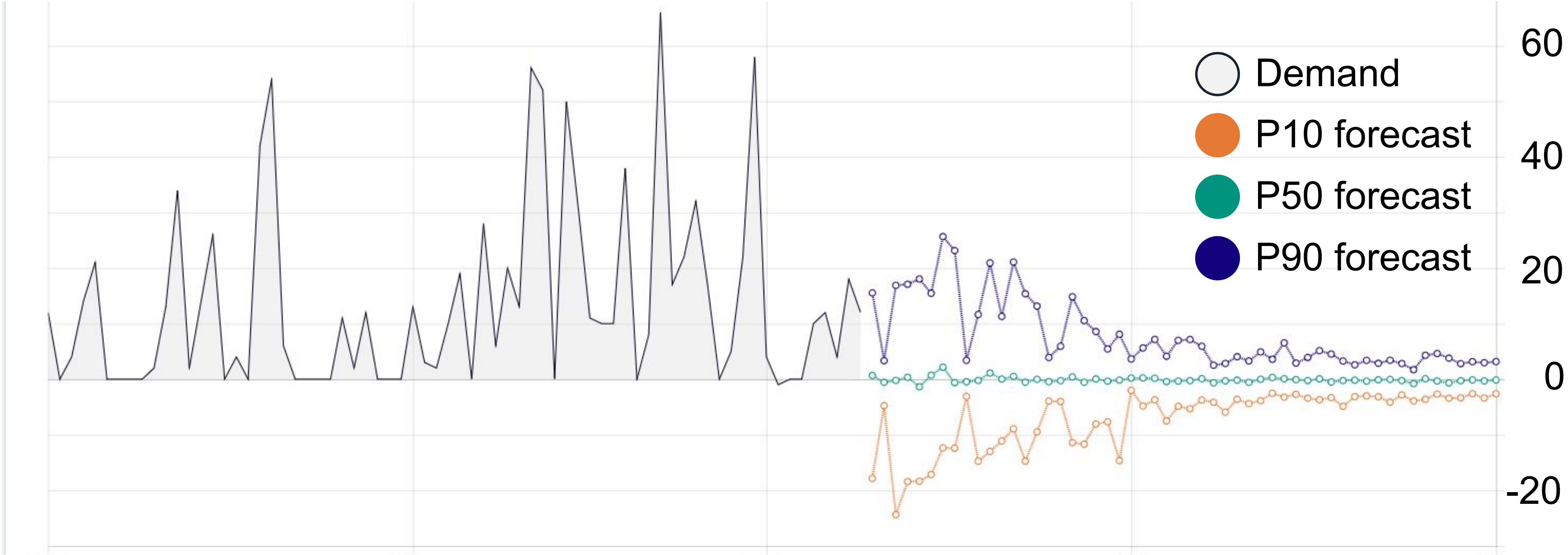
Evaluating your forecast: Back testing

Predictor accuracy metrics are based on back testing.



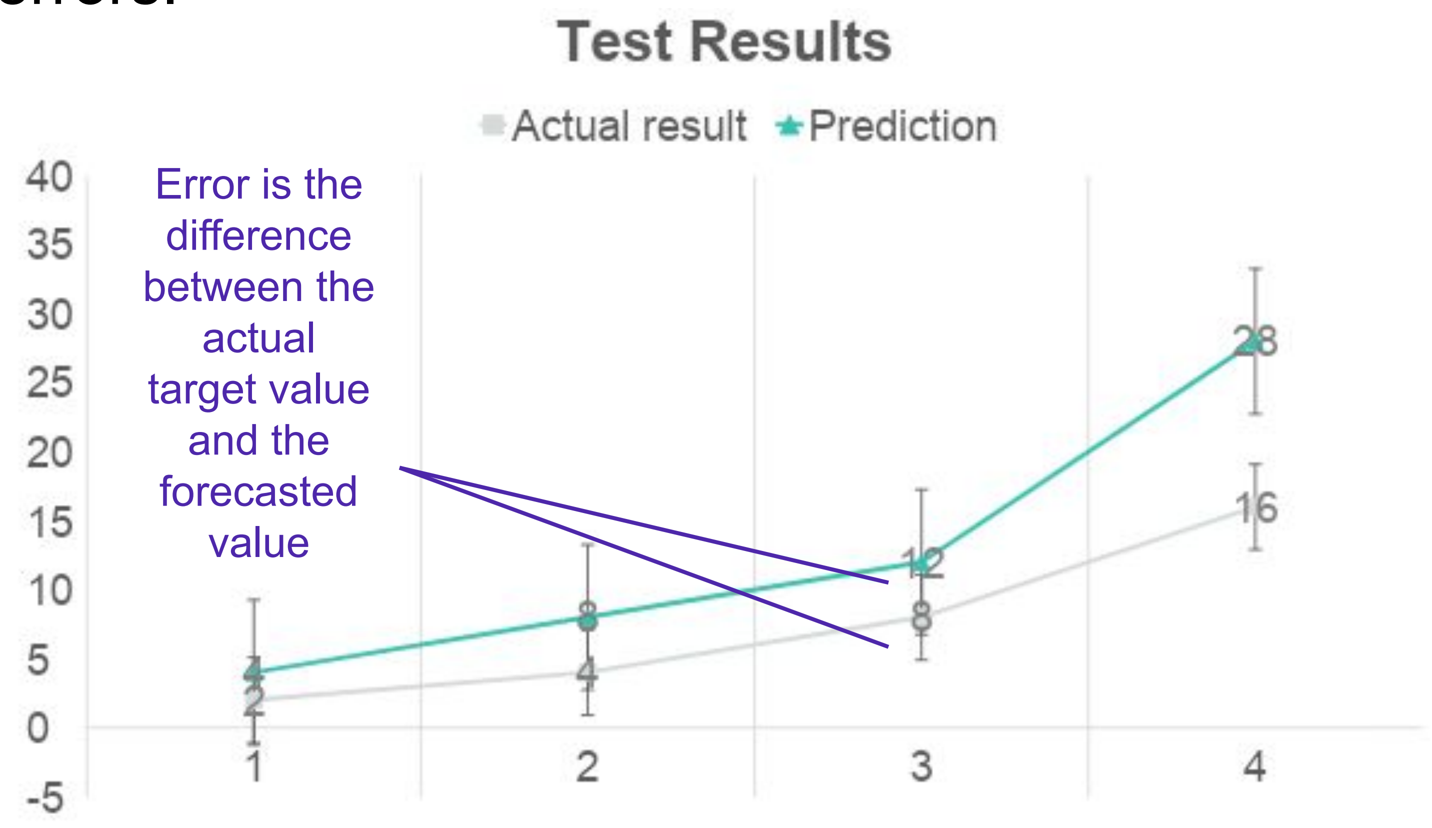
Evaluation metrics: wQuantileLoss

- Quantiles determined for 10%, 50%, and 90% quantiles
- wQuantileLoss is the average error for each quantile in a set –
  - Works best for models with greater variability in the errors



Root mean square error (RMSE)

RMSE is the square of the errors.



Test	Actual result	Prediction	Deltas
1	2	4	2
2	4	8	4
3	8	12	4
4	16	28	12
		RMSE	6.708204

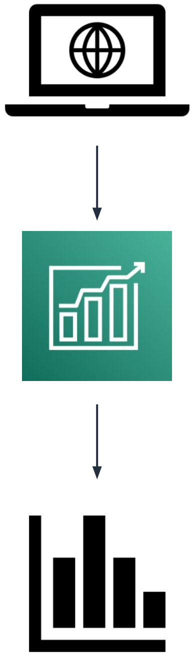
Works best for a model where most errors are of similar size

Model accuracy example

Web retailer of shoes wants to predict how often it will be unable to fill orders for AnyCompany brand shoes.

Amazon Forecast predicts demand of 1,000 pairs per month

- P10: 10% of the time, fewer than 880 pairs will be ordered
- P50: 50% of the time, fewer than 1,050 pairs will be ordered
- P90: 90% of the time, fewer than 1,200 pairs will be ordered



P10 = 880
P50 = 1050
P90 = 1200

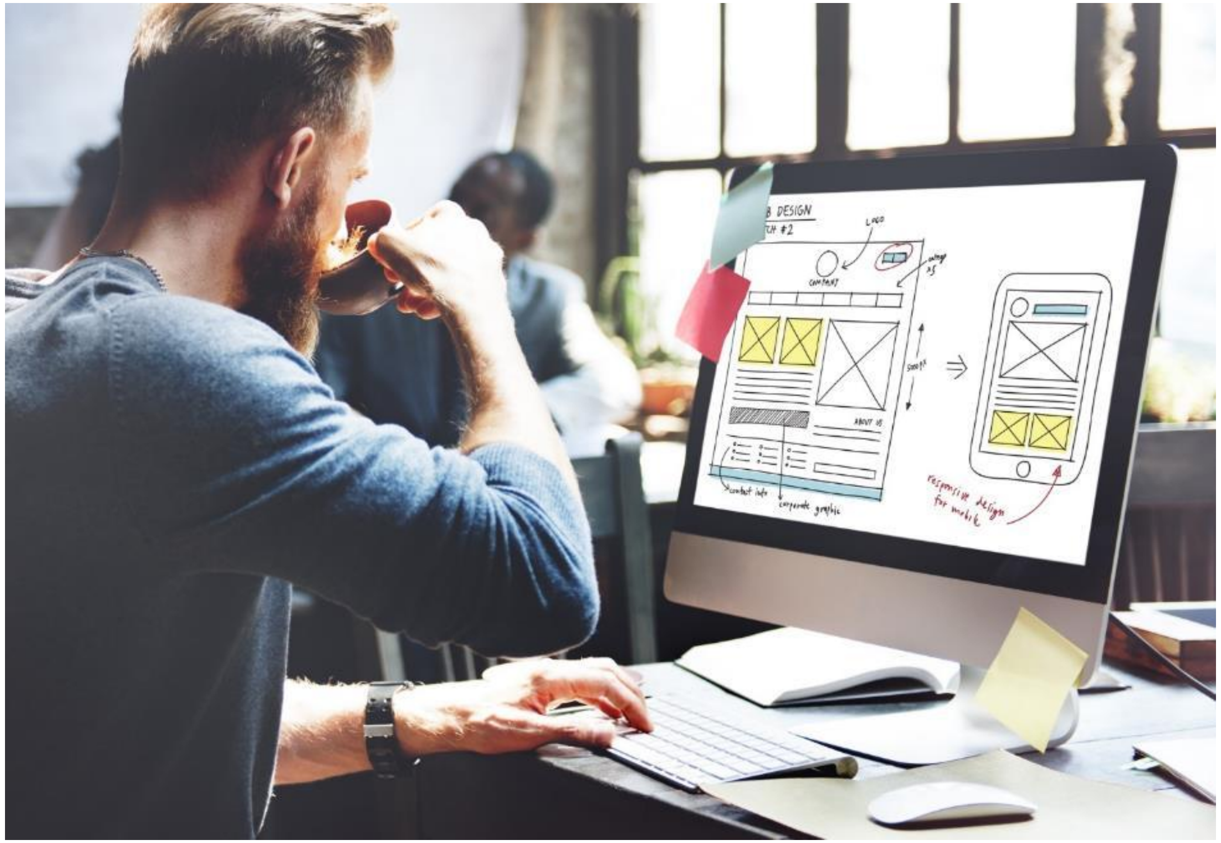
Forecast = 1000

Demonstration: Creating a forecast with Amazon Forecast

Section 3 key takeaways

- You can use Amazon Forecast for time series data
- Schemas are specific for domains
- Data can include –
  - Time series data
  - Metadata
  - Related data
- Data is split into training and testing data by accounting for time
- Use RMSE and wQuantileLoss metrics to evaluate model

Module 4 – Guided Lab: Creating a Forecast with Amazon Forecast



Module 4: Introducing Forecasting

Module wrap-up

Module summary

In summary, in this module you learned how to:

- Describe the business problems solved by using Amazon Forecast
- Describe the challenges of working with time series data
- List the steps that are required to create a forecast by using Amazon Forecast
- Use Amazon Forecast to make a prediction

Complete the knowledge check

Additional resources

- Amazon Forecast documentation
- Amazon Forecast product page
- How to not use machine learning for time series forecasting
- Time series forecasting principles Amazon Forecast whitepaper

Thank you