AWS Academy Machine Learning Foundations
Module 04 Student Guide
Version 1.0.3

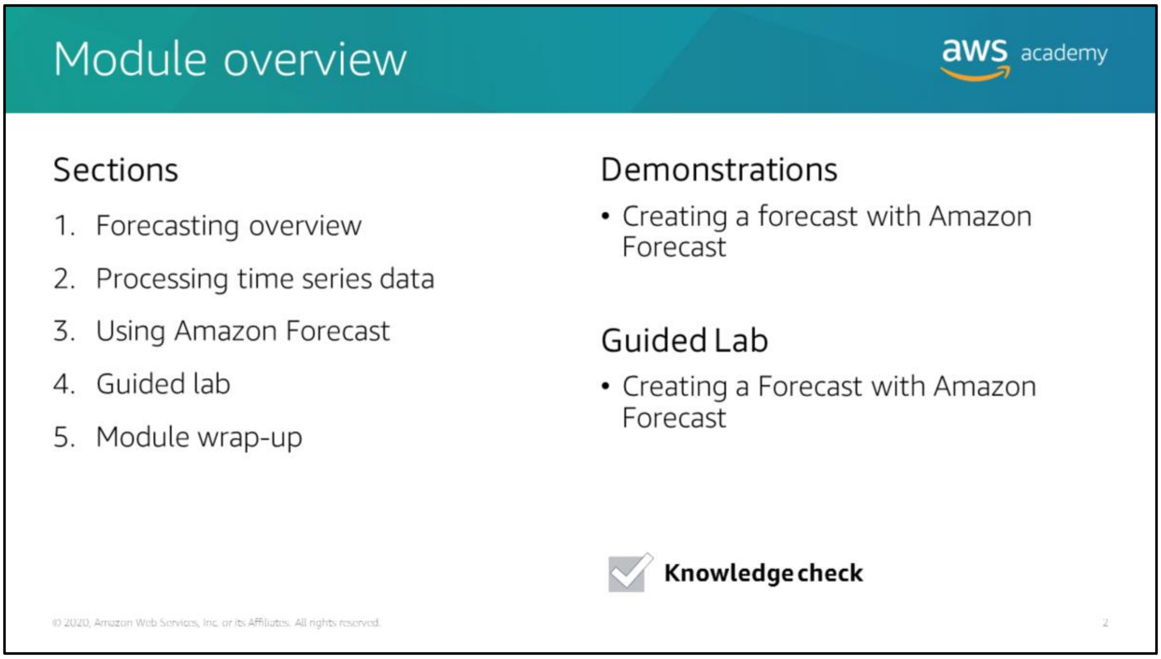
200-ACMLFO-10-EN-SG

Contents

AWS Academy Machine Learning Foundations

Module 4: Introducing Forecasting

Welcome to Module 4: Introduction to Forecasting.



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

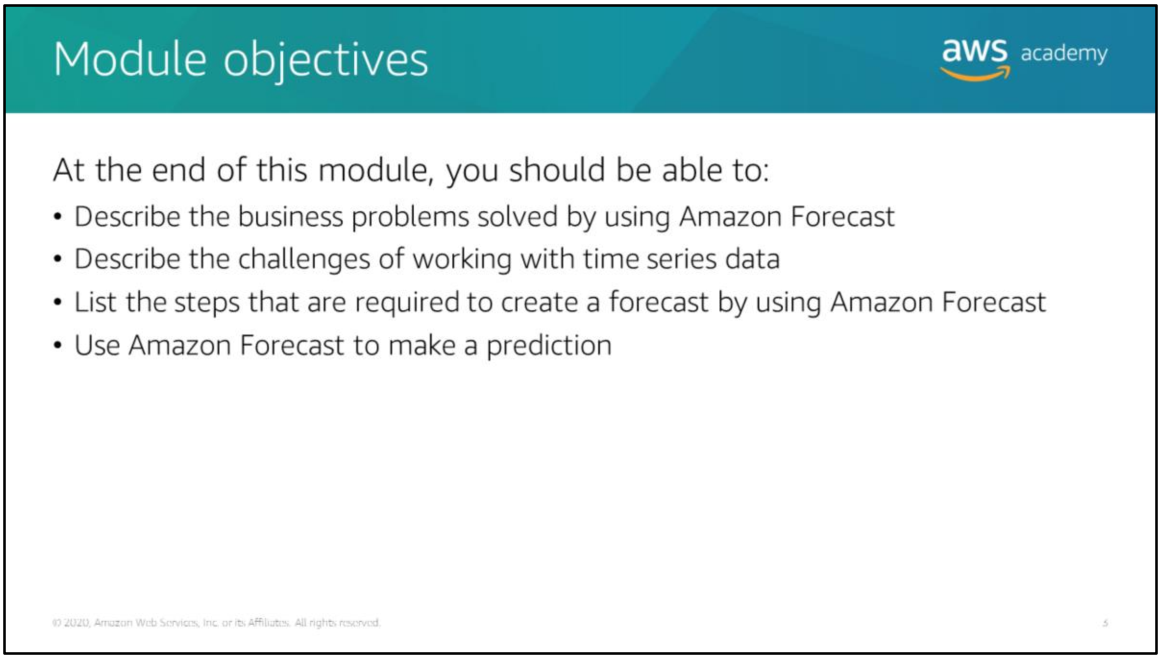
☑ **Knowledge check**

This module includes the following topics:

- Forecasting overview
- Processing time series data
- Using Amazon Forecast
- Guided lab
- Module wrap-up

The module also includes a hands-on guided lab where you will learn how to use Amazon Forecast when you work with time series data.

Finally, you will be asked to complete a knowledge check that will test your understanding of key concepts covered in this module.



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

After completing this module, you should be able to:

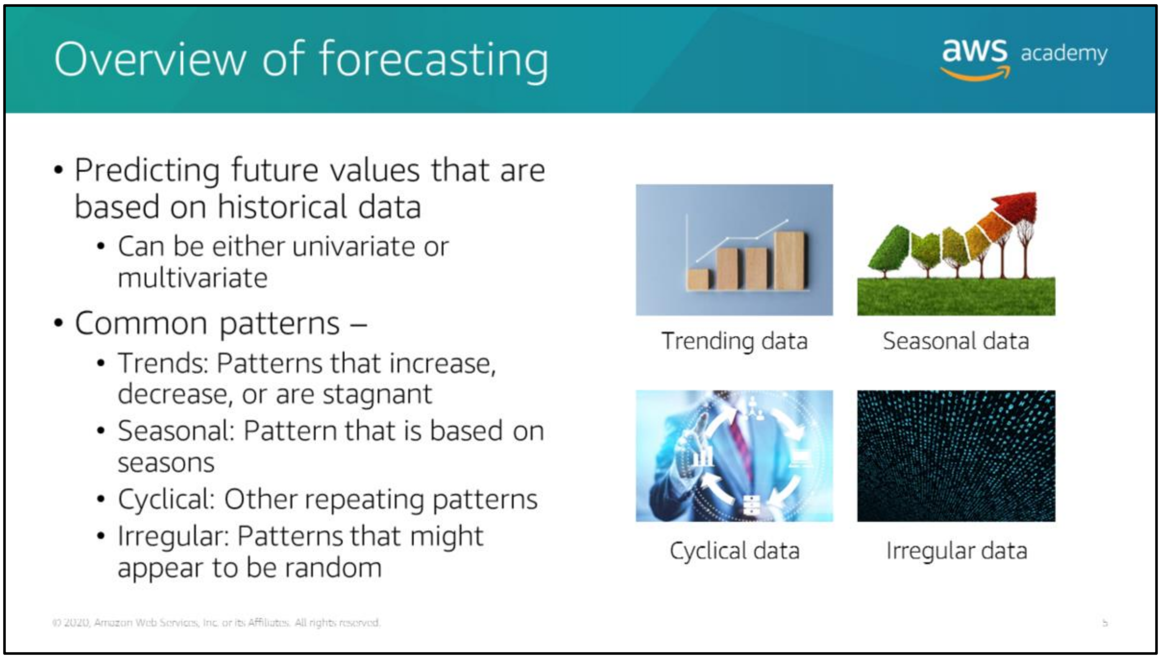
- Describe the business problems solved by using Amazon Forecast
- Describe the challenges of working with time series data
- List the steps that are required to create a forecast by using Amazon Forecast
- Use Amazon Forecast to make a prediction

Module 4: Introducing Forecasting

Section 1: Forecasting overview

Introducing Section 1: Forecasting overview.

You start with a review of what forecasting means and learn about some use cases for forecasting.



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

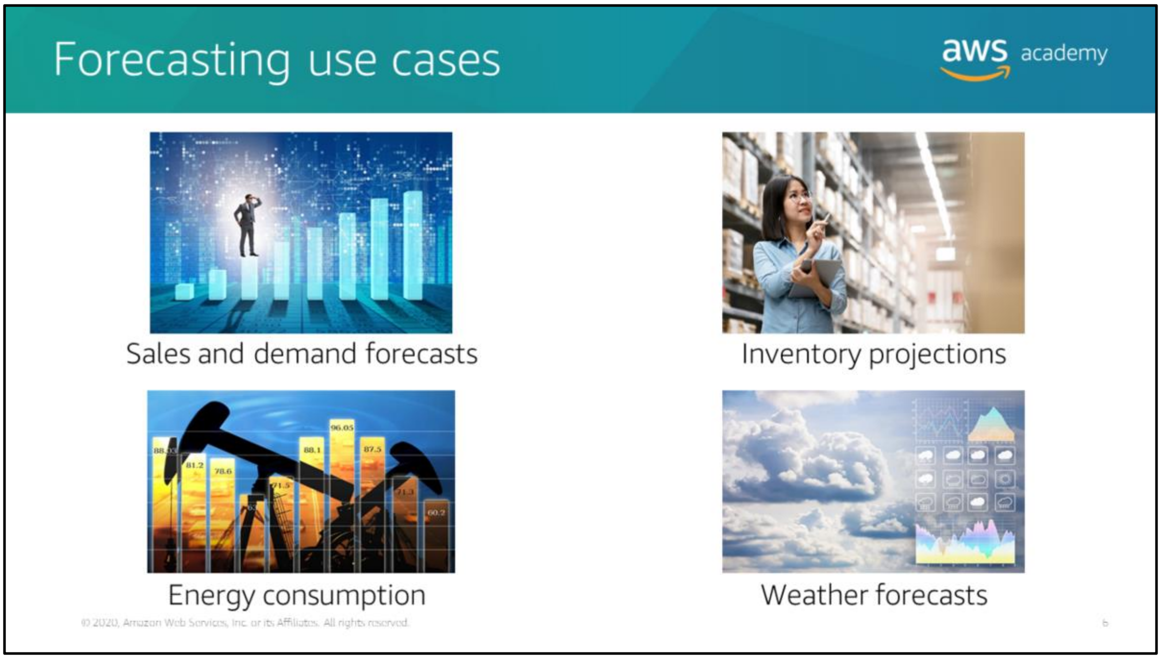
Cyclical data

Irregular data

Forecasting is an important area of machine learning. It is important because so many opportunities for predicting future outcomes are based on historical data. Many of these opportunities involve a time component. Although the time component adds more information, it also makes time series problems more difficult to handle than other types of predictions.

You can think of time series data as falling into two broad categories. The first type is *univariate*, which means that it has only one variable. The second type is *multivariate*, which means that it has more than one variable. In addition to these two categories, most time series datasets also follow one of the following patterns:

- Trend – A pattern that shows the values as they increase, decrease, or stay the same over time
- Seasonal – A repeating pattern that is based on the seasons in a year
- Cyclical – Some other form of a repeating pattern
- Irregular – Changes in the data over time that appear to be random or that have no discernable pattern



Forecasting use cases

Sales and demand forecasts

Inventory projections

Energy consumption

Weather forecasts

You can use forecasting for a range of domains. Some of the more common applications include:

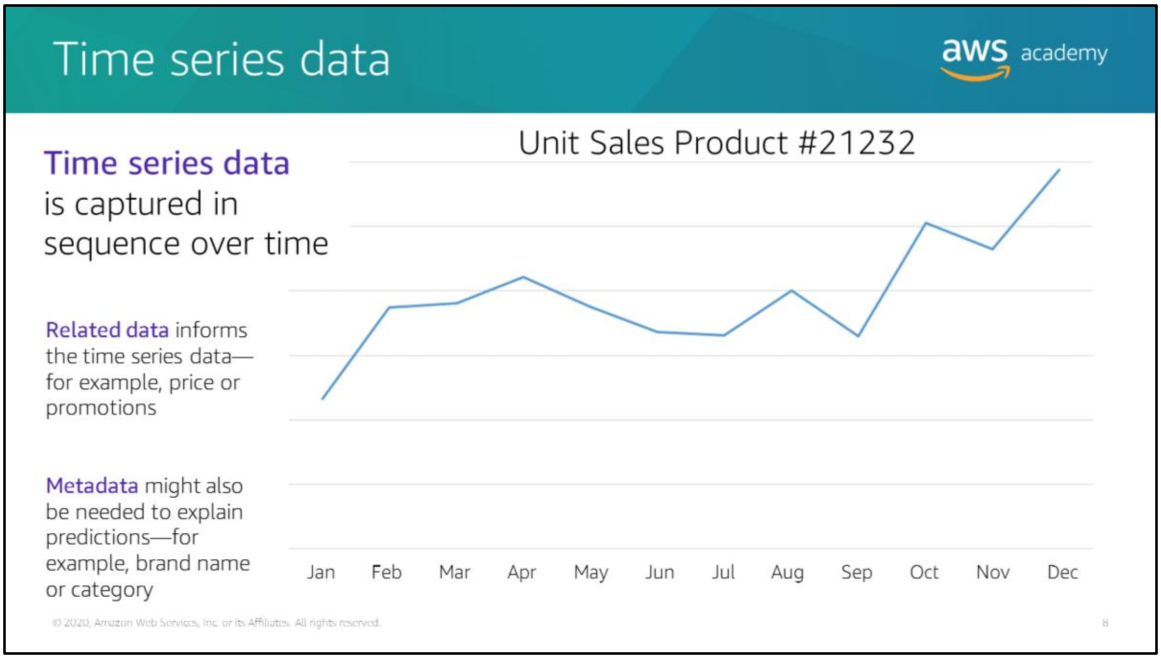
- Marketing applications, such as sales forecasting or demand projections.
- Inventory management systems to anticipate required inventory levels. Often, this type of forecast includes information about delivery times.
- Energy consumption to determine when and where energy is needed.
- Weather forecasting systems for governments, and commercial applications such as agriculture.

Module 4: Introducing Forecasting

Section 2: Processing time series data

Introducing Section 2: Processing time series data.

Working with time series data presents several unique challenges, which you will now review.



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

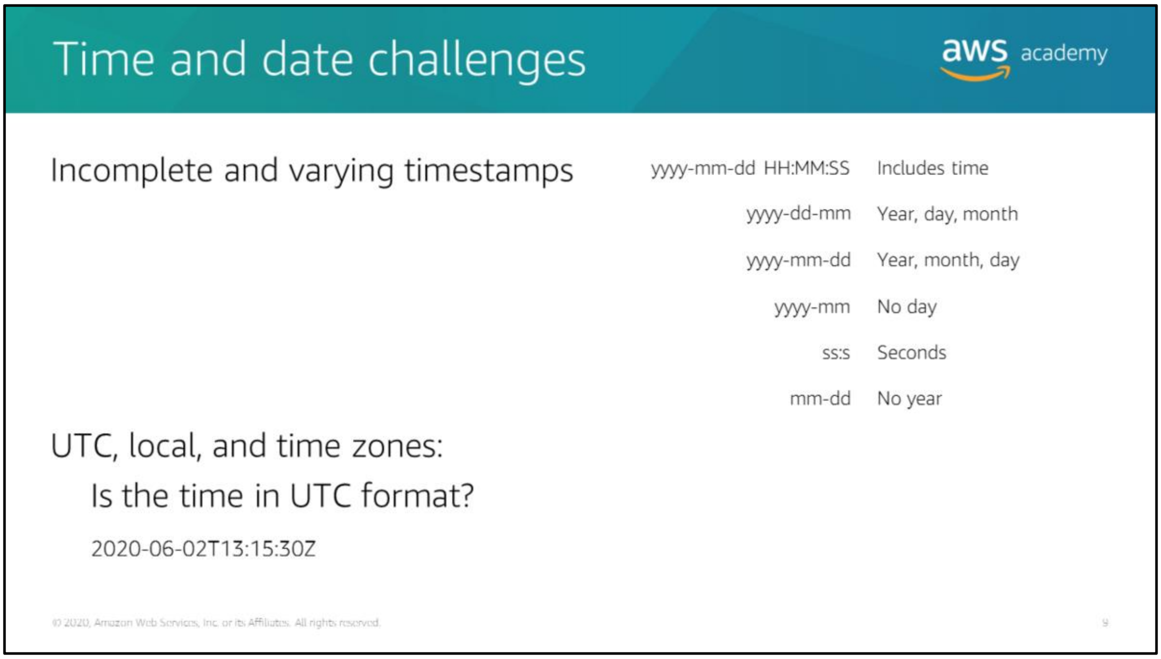
8

Time series data is captured in chronological sequence over a defined period of time. Introducing time into a machine learning model has a positive impact because the model can derive meaning from change in the data points over time. Time series data tends to be correlated, which means that a dependency exists between data points.

Because you have a regression problem—and because regression assumes independence of data points—you must develop a method for handling data dependence. The purpose of this method is to increase the validity of the predictions.

In addition to the time series data, you can add related data to augment a forecasting model. For example, for a prediction about retail sales, you might include information about the product being sold (such as item identification or sales price). This information is in addition to the number of units that are sold per time period.

The third type of data is metadata about the dataset. For example, for a retail dataset, you might want to include metadata to group results, like a brand name. Another example of metadata could be including a genre for music or videos.



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

9

The more data that you have, the better. A challenge that you can expect with multiple data sources is the timestamp of the data. You find differences in the timestamp format, along with other challenges, such as incomplete data. You might be able to infer missing data in some cases. For example, imagine that you have some data that contains both the month and the day, but no year. Suppose that the data appears to sequence through the month numbers in the database, and repeats after 12. In that case, you can add the year if you know when the data started. You can infer future years, based on the order of the data.

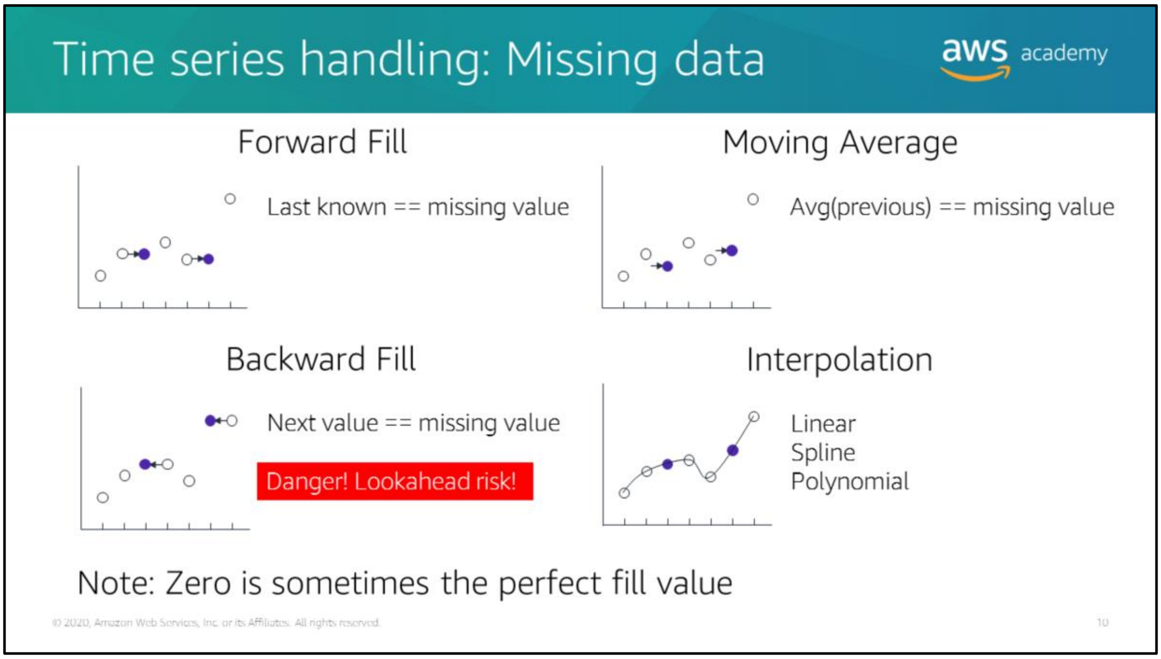
Much data is stored in Universal Coordinated Time (UTC) format, but not all data is in UTC. You should check whether the timestamp is local or universal time.

Sometimes the timestamp doesn’t represent the time that you think it does. For example, suppose you have a database of cars that were serviced at a garage. Does the timestamp indicate the time that the car arrived, was completed, or was picked up? Or does it indicate when the final entry was entered into the system?

If you try to model hourly caloric intake of patients, but you have only daily data, then you must adjust your target timescale.

You might not have a timestamp present in your data. You might have other ways to extrapolate a time series, depending on the data and domain. For example, you might have wavelength measurements or vectors in an image.

Daylight savings time is different around the world. Because of daylight savings, some times occur twice a year in their time zones.



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 risk!

Interpolation

Linear
Spline
Polynomial

Note: Zero is sometimes the perfect fill value

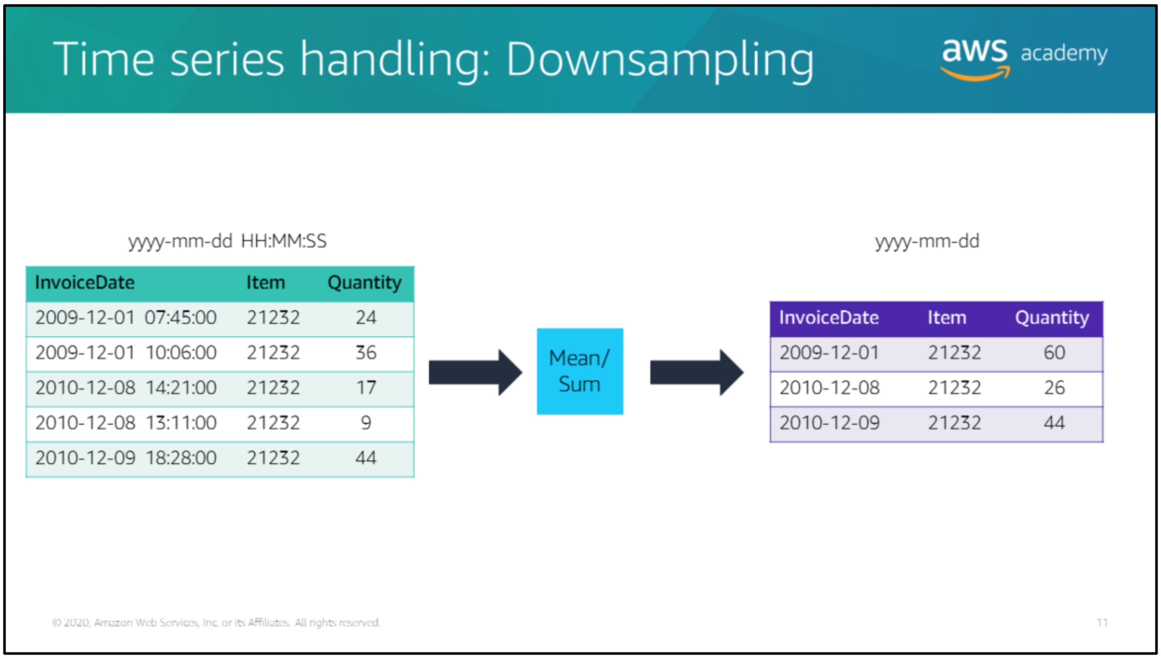
A common occurrence in real-world forecasting problems is missing values in the raw data. Missing values makes it harder for a model to generate a forecast. The primary example in retail is an out-of-stock situation in demand forecasting. If an item goes out of stock, the sales for the day will zero. If the forecast is generated based on those zero sales values, the forecast will be incorrect.

Missing values can be marked as missing for various reasons. Missing values can occur because of no transaction, or possibly because of measurement errors. Maybe a service that monitored certain data was not working correctly, or the measurement could not occur correctly.

The missing data can be calculated in several ways:

- Forward fill – Uses the last known value for the missing value.
- Moving average – Uses the average of the last known values to calculate the missing value.
- Backward fill – Uses the next known value after the missing value. Be aware that it is a potential danger to use the future to calculate the past, which is bad in forecasting. This practice is known as *lookahead*, and it should be avoided.
- Interpolation – Essentially uses an equation to calculate the missing value.

You also have the choice to use a zero fill. This choice is often used in retail, a domain where missing sales data shouldn’t be calculated. The missing data represents no orders on that day. It would be wise to investigate why, but you don’t want to fill in the missing value in this case.



Time series handling: Downsampling

yyyy-mm-dd HH:MM:SS

InvoiceDate	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

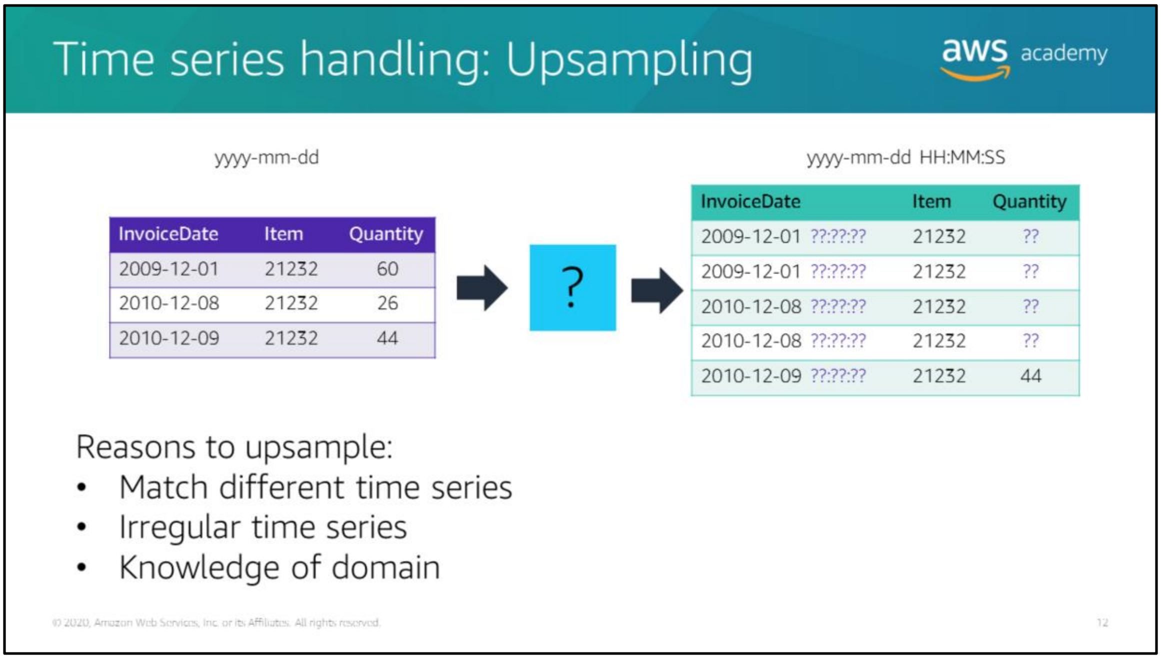
yyyy-mm-dd

InvoiceDate	Item	Quantity
2009-12-01	21232	60
2010-12-08	21232	26
2010-12-09	21232	44

You might obtain data at different frequencies. For example, you might have sales data that includes the exact timestamp that the sale was recorded. However, the inventory data might contain only the year, month, and day of the inventory level. When you have data that is at a different frequency than other datasets, or isn’t compatible with your question, you might need to *downsample*.

Downsample means moving from a more finely grained time to a less finely grained time. This example converts an hourly dataset to a daily dataset.

When you downsample, you must decide how to combine the values. In the case of sales data, summing the quantity makes the most sense. If the data is temperature, you might want to find the average. Understanding your data helps you decide what the best course of action is.



Time series handling: Upsampling

yyyy-mm-dd

InvoiceDate	Item	Quantity
2009-12-01	21232	60
2010-12-08	21232	26
2010-12-09	21232	44

?

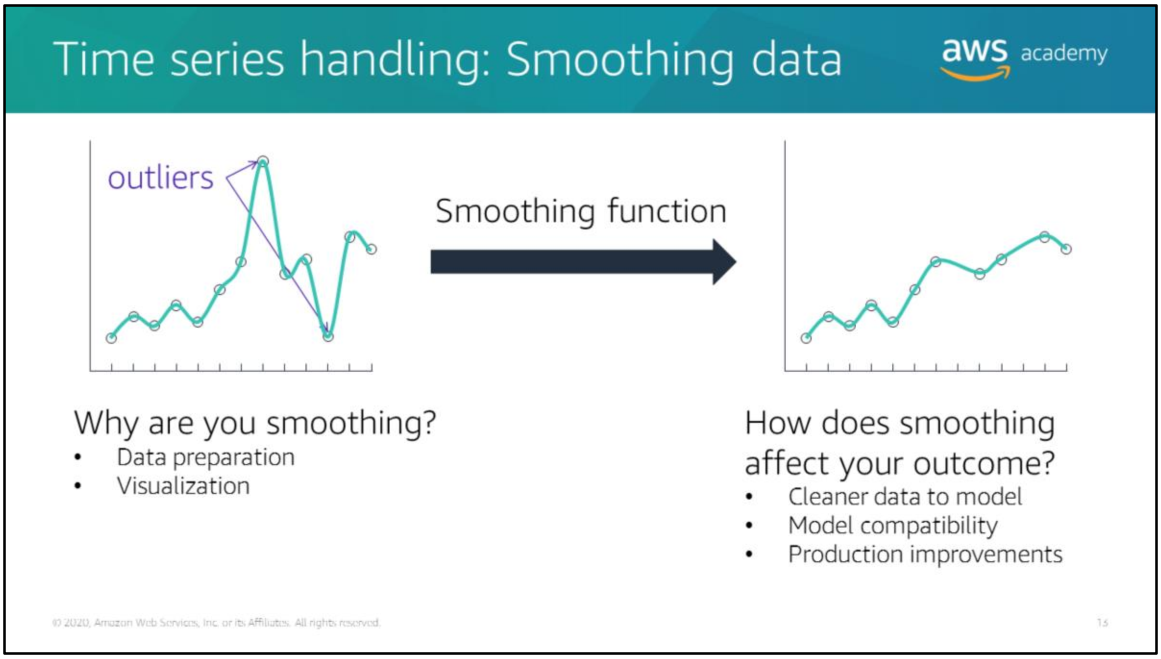
yyyy-mm-dd HH:MM:SS

InvoiceDate	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

The inverse of downsampling is upsampling. The problem with upsampling is that it’s difficult to achieve in most cases. Suppose that you wanted to upsample your sales data from daily sales to hourly sales. Unless you have some other data source to reference, you wouldn’t be able to change from daily to hourly sales. In some cases, you must use additional data or knowledge. For example, if you must match the frequency of another time series, you might have an irregular time series or specific domain knowledge that could help. In those cases, you must be careful of how you convert the data. For the retail example, the best that you can do is create a single order for the day at a specified hour. For temperature, you can copy the daily temperature into each of the hourly slots, or use some formula to calculate a curve.



Time series handling: Smoothing data

aws academy
outliers

Smoothing function

Why are you smoothing?

- Data preparation
- Visualization

How does smoothing affect your outcome?

- Cleaner data to model
- Model compatibility
- Production improvements

15

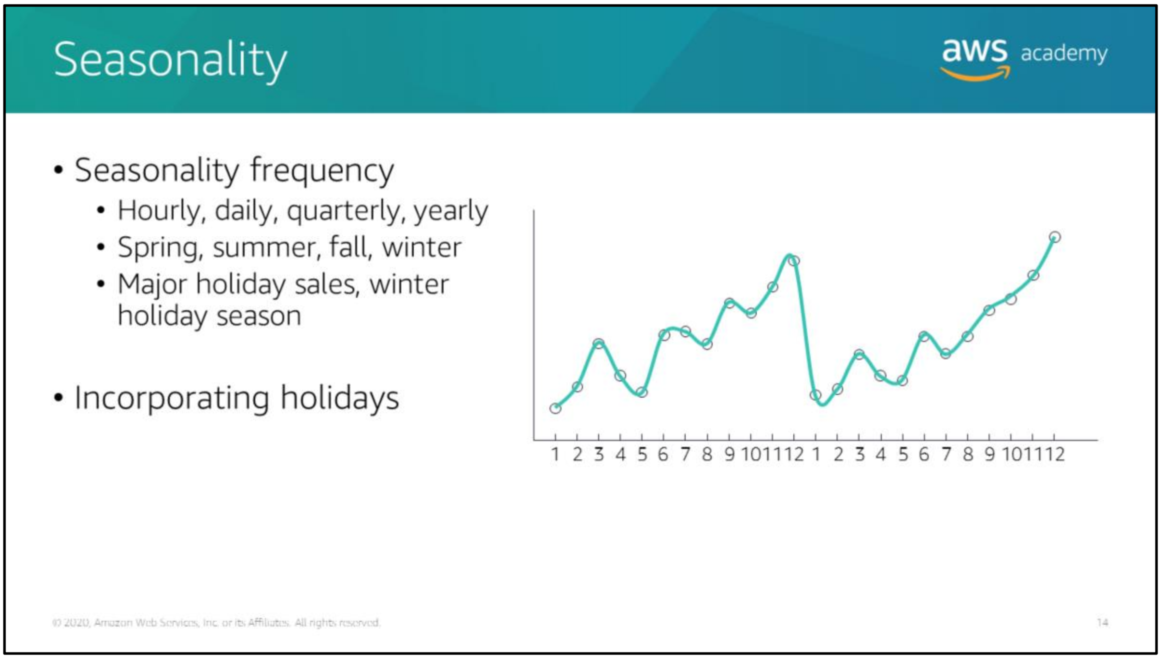
Outliers can be a problem in data science. The same is true of time series data.

If you examine sales data and you see an order with an unusually large quantity of items, you might not want to include that order in your forecast calculations. The order size might never be repeated. Removing these outliers and anomalies is known as *smoothing*.

Smoothing your data can help you deal with outliers and other anomalies. You might consider smoothing for the following reasons.

- Data preparation – Removing error values and outliers
- Visualization – Reducing noise in a plot

Understand why you are smoothing the data and the impact that it might have. You might want the outcome to be reduced noise and to create a better model. However, it’s equally important to consider these questions: Could your smoothing compromise the model? Does the model expect noisy data? Can you also smooth the data in production?



Seasonality

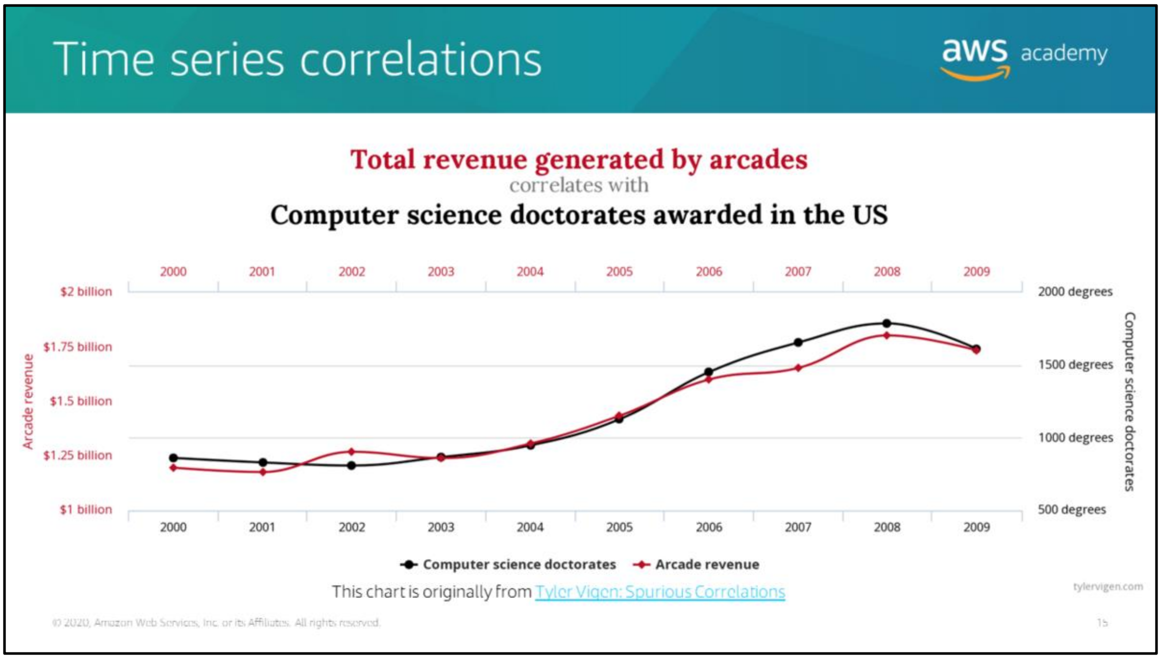
aws academy

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

14

Seasonality in data is any kind of repeating observation where the frequency of the observation is stable. For example, in sales you typically see higher sales at the end of a quarter, and into Q4. Consumer retail exhibits this pattern even more in Q4. Be aware that data can have multiple types of seasonality in the same dataset.

Many times, you might want to incorporate seasonality information into your forecast. Localized holidays are a good example for sales.



Time series correlations

Total revenue generated by arcades
correlates with
Computer science doctorates awarded in the US

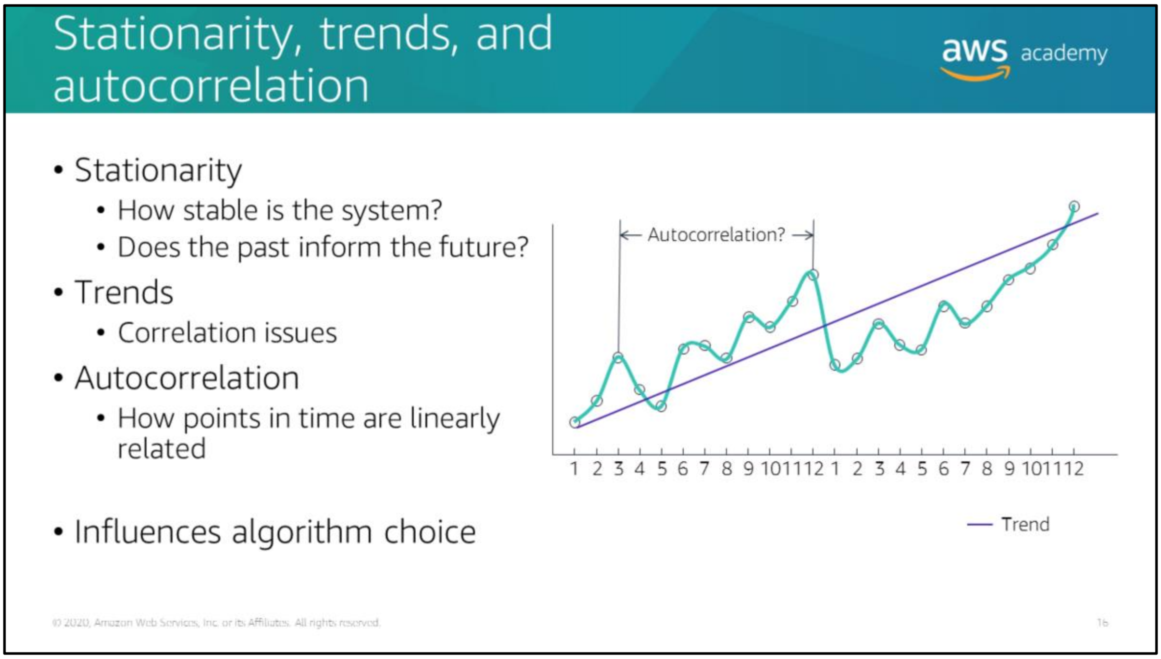
This chart is originally from Tyler Vigen: Spurious Correlations

15

Correlations do not mean causation.

Be careful when you interpret your own data, and be cautious about correlations—you don't want to act on correlations that have no real-world meaning. As an experiment, say that you generate two random time series datasets of numbers between 0 and 1. You will find that they have low correlation. However, if you introduce the same slope to both sets of data, you will see a strong correlation.

For more correlations, see Spurious correlations. Many correlations are plotted, and none of them make any sense.



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

16

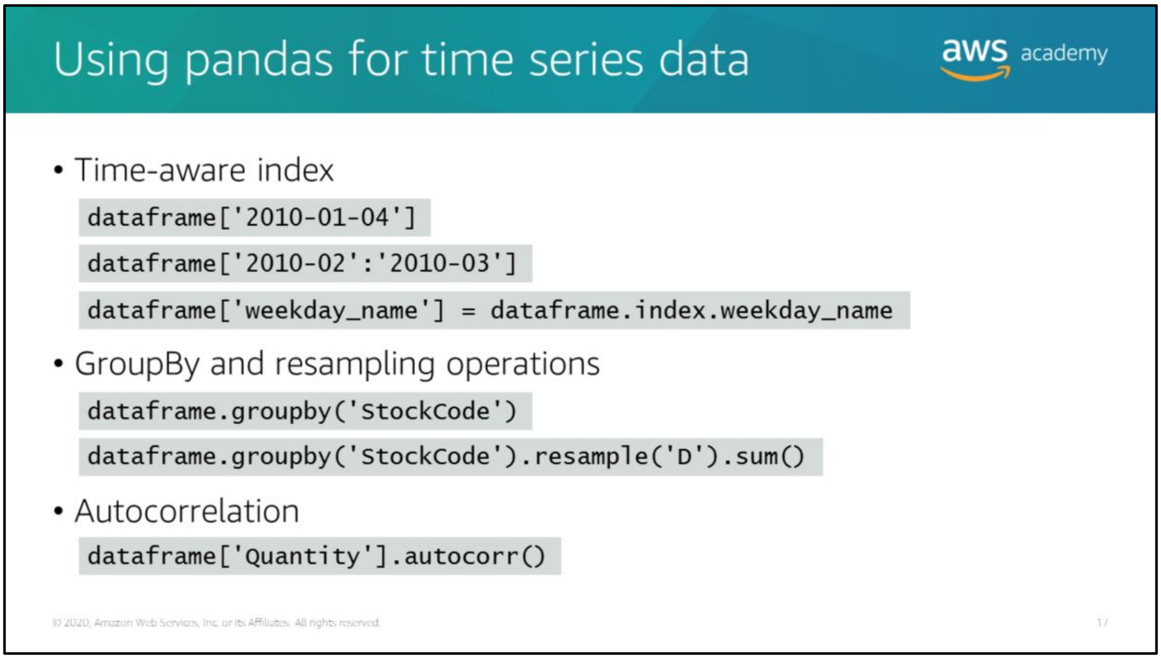
It is important to know how stable a system is. The level of stability, or *stationarity*, can tell you how much you should expect the system's past behavior to inform future behavior. A system with low stability is not good for predicting the future.

Often, you will want to determine the trend for a time series. However, adjusting the series for the trend can make it difficult to compare the series with another series that was also adjusted for trend. The trends might dominate the values in the series, which can lead you to overestimate of the correlation between the two series. This phenomenon was shown in the previous topic. For a detailed explanation of reasons for this occurrence, see Avoiding Commons Mistakes with Time Series Analysis

Autocorrelation is one of the special problems that you face with time series data. As you saw in other machine learning problems, the goal of building an ML model is to separate the signal from the noise. Autocorrelation is a form of noise because separate observations are not independent of each other.

A time series with autocorrelation might overstate the accuracy of the model that is produced. Some of the algorithms that you see in this module can help correct for autocorrelation.

These factors, along with seasonality, can influence the model that you select to produce your forecast. Some algorithms handle seasonality and autocorrelation, but others do not.



Using pandas for time series data

aws academy

- 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()
```

17

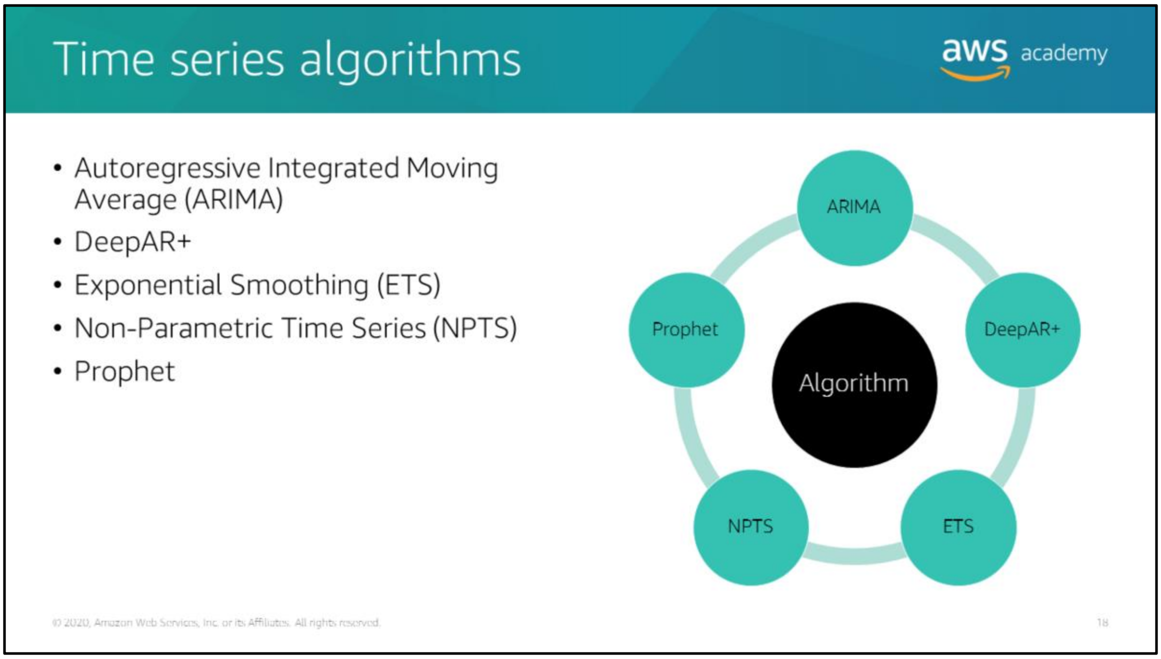
The pandas library was developed with financial data analysis in mind. As such, it is good at handling time series data.

You can set the index of your pandas DataFrame to be a *datetime*, which enables you to use the date and time to select your data. You can use ranges that contain partial dates. You can also extract date parts, such as *year*, *month*, *weekday_name*, and more.

For grouping and resampling tasks, pandas has built-in functions to do both.

Finally, pandas can give you insights into autocorrelation.

For more information about pandas and time series data, see Time series / date functionality In the pandas documentation..



Time series algorithms

aws academy

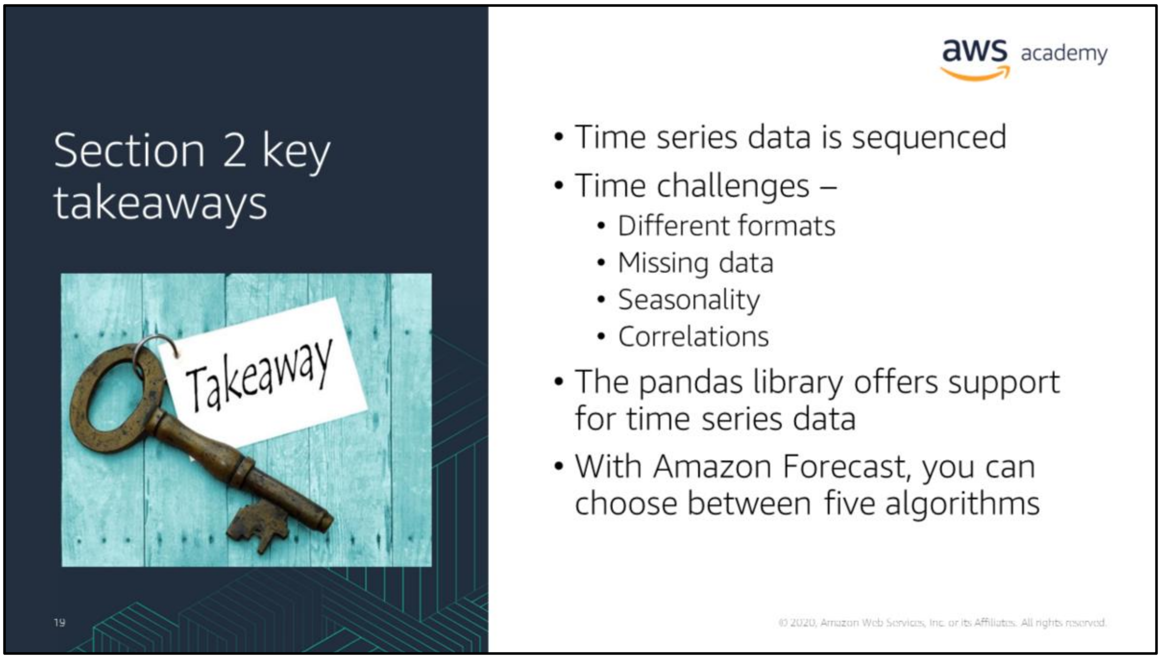
- Autoregressive Integrated Moving Average (ARIMA)
- DeepAR+
- Exponential Smoothing (ETS)
- Non-Parametric Time Series (NPTS)
- Prophet

18

One of the tasks in building a forecasting application is to choose an appropriate algorithm. The type of dataset that you are using and the features of that dataset should determine your choice of algorithm.

Amazon Forecast supports these five algorithms.

- Autoregressive Integrated Moving Average (ARIMA) – This algorithm removes autocorrelations, which might influence the pattern of observations.
- DeepAR+ – A supervised learning algorithm for forecasting one-dimensional time series. It uses a recurrent neural network to train a model over multiple time series.
- Exponential Smoothing (ETS): This algorithm is useful for datasets with seasonality. It uses a weighted average for all observations. The weights are decreased over time.
- Non-Parametric Time Series (NPTS) – Predictions are based on sampling from past observations. Specialized versions are available for seasonal and climatological datasets.
- Prophet – A Bayesian time series model. It's useful for datasets that span a long time period, have missing data, or have large outliers.



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

19

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

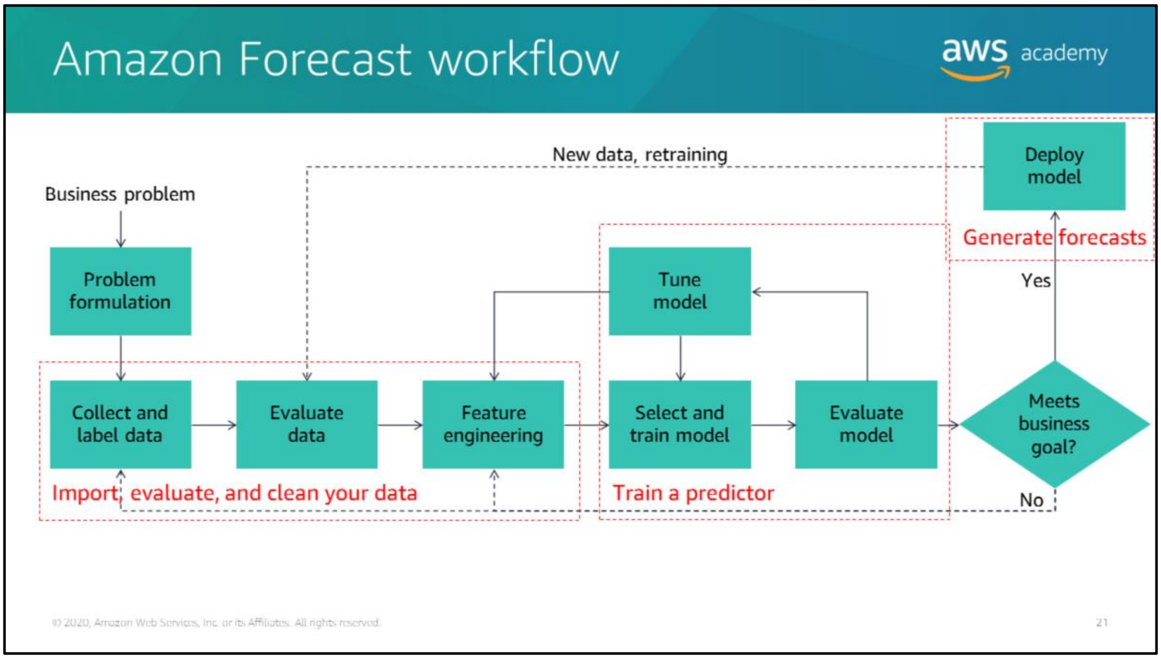
Some key takeaways from this section of the module include:

- Time series data is sequenced data that includes a time element, which makes it different from regular datasets
- Some of the time challenges include –
  - Handling different time formats
  - Handling missing data through down sampling, up sampling and smoothing
  - Handling seasonality, such as weekdays and yearly cycles
  - Avoiding bad correlations
- The pandas library offers support for time series data through functions that deal with time
- With Amazon Forecast, you can choose between five algorithms –
  - ARIMA
  - DeepAR+
  - ETS
  - NPTS
  - Prophet

Module 4: Introducing Forecasting

Section 3: Using Amazon Forecast

Introducing Section 3: Using Amazon Forecast.



Amazon Forecast workflow

aws academy

Business problem

Problem formulation

Collect and label data

Evaluate data

Feature engineering

Import, evaluate, and clean your data

New data, retraining

Tune model

Select and train model

Evaluate model

Train a predictor

Meets business goal?

Yes

No

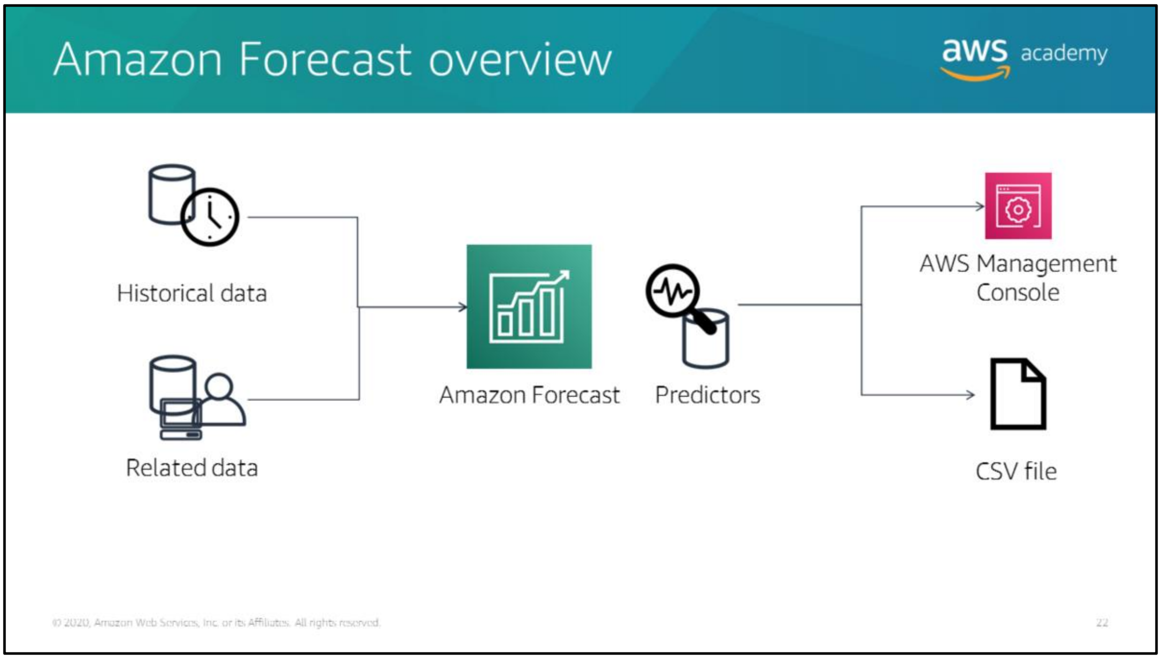
Deploy model

Generate forecasts

21

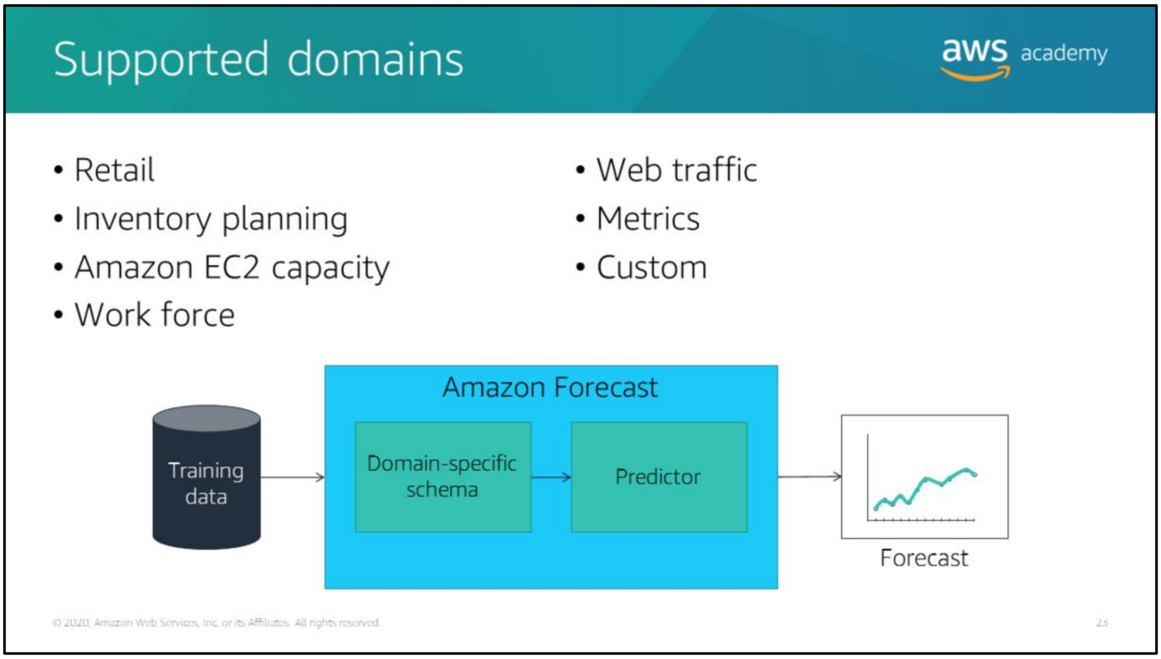
When you generate forecasts, you can apply the machine learning development pipeline that you use throughout this course.

- Import your data – You must import as much data as you have—both historical data and related data. You should do some basic evaluation and feature engineering before you use the data to train a model.
- Train a predictor – To train a predictor, you must choose an algorithm. If you are not sure which algorithm is best for your data, you can let Amazon Forecast choose by selecting *AutoML* as your algorithm. You also must select a domain for your data, but if you're not sure which domain fits best, you can select a custom domain. Domains have specific types of data that they require. For more information, see Predefined Dataset Domains and Dataset Types in the Amazon Forecast documentation.
- Generate forecasts – As soon as you have a trained model, you can use the model to make a forecast by using an input dataset group. After you generate a forecast, you can query the forecast, or you can export it to an Amazon Simple Storage Service (Amazon S3) bucket. You also have the option to encrypt the data in the forecast before you export it.



Amazon Forecast overview

The overall process for working with Amazon Forecast is to import historical and related data. Amazon Forecast inspects the data, identifies key data, and selects an appropriate algorithm. It uses the algorithm to train and optimize a custom model and produce a predictor. You create forecasts by applying the predictor to your dataset. Then, you can either retrieve these forecasts in the AWS console, or export the forecasts as comma-delimited files. You can also use an application programming interface (API) and command line interface (CLI) commands to create and retrieve forecasts.



Supported domains

- Retail
- Inventory planning
- Amazon EC2 capacity
- Work force
- Web traffic
- Metrics
- Custom

Training data

Amazon Forecast

Domain-specific schema

Predictor

Forecast

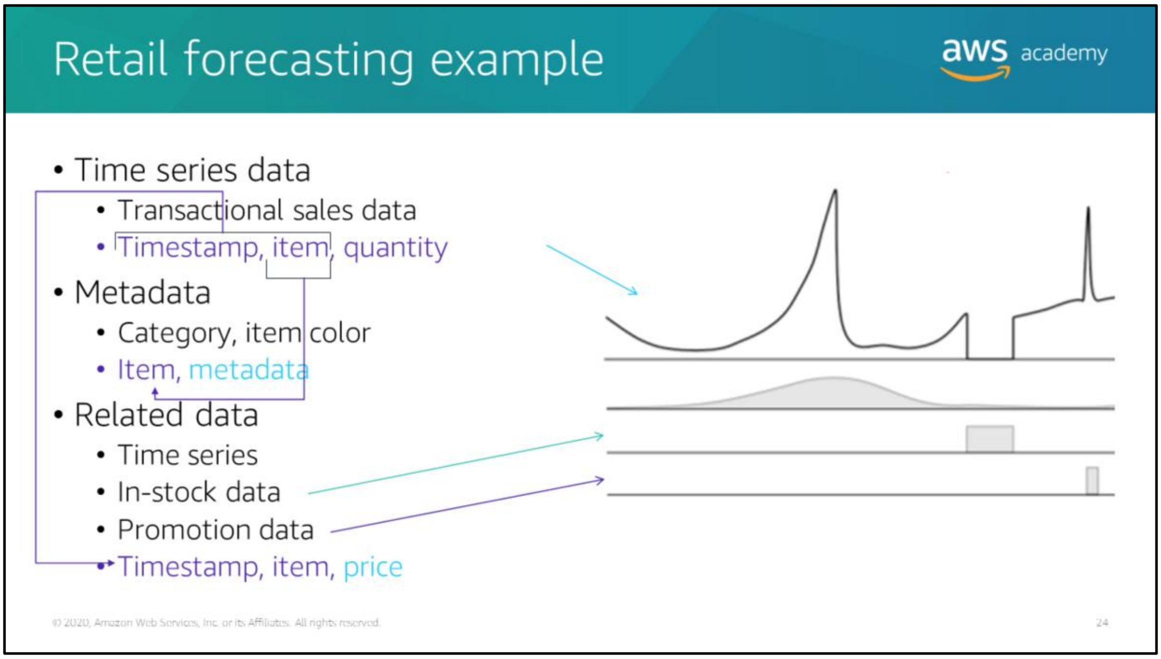
23

When you work with Amazon Forecast, you should select the appropriate domain. You can select from the following list:

- Retail – Product demand
- Inventory planning – Raw materials requirements
- EC2 capacity – Capacity demand for Amazon Elastic Compute Cloud (Amazon EC2)
- Work force – Workload projections
- Web traffic – Projected traffic to one or more websites
- Metrics – Projecting metrics such as revenue, sales, or cash flow
- Custom – Projections for a domain that you can't map to one of the previous domains

By selecting a domain, you improve the efficiency of the predictor. Each domain has specific types of data that you supply when you build the predictor. For example, the *Retail* domain expects item identifiers, a timestamp for the observation, the number of sales for that item, and the specified timestamp.

For more information on the supported domains, see the topic Predefined Dataset Domains and Dataset Types



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

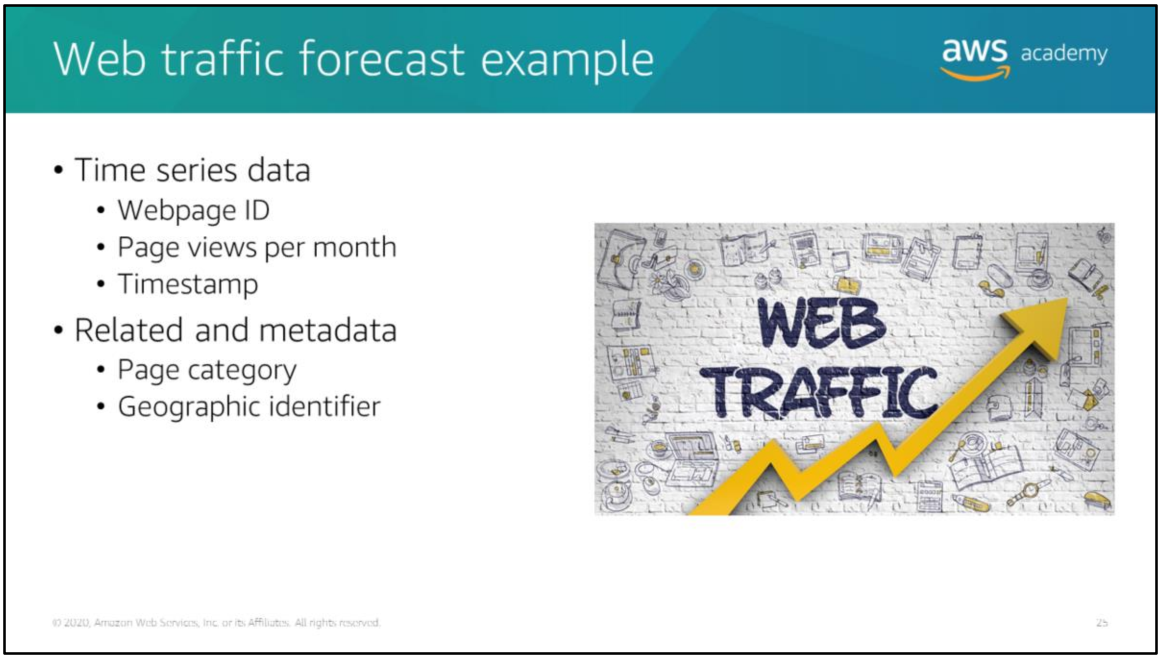
The following example shows the data that you need for a retail demand forecast.

For the time series, you must provide:

- Timestamp – The time at which the transaction took place, ideally in UTC format
- Item – The item ID of the item
- Quantity – How many items were sold

The metadata for the item might include category or item color, for example. The link back to the time series data includes only the item ID, because the metadata for the item doesn't typically change.

The sales price or other promotion data are examples of related data that can create a more useful forecast. To link this back to the item, you must include the timestamp and item ID.



Web traffic forecast example

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

aws academy
WEB
TRAFFIC

25

The following example shows the kind of data that you need for a web traffic forecast.

For the time series, you must provide:

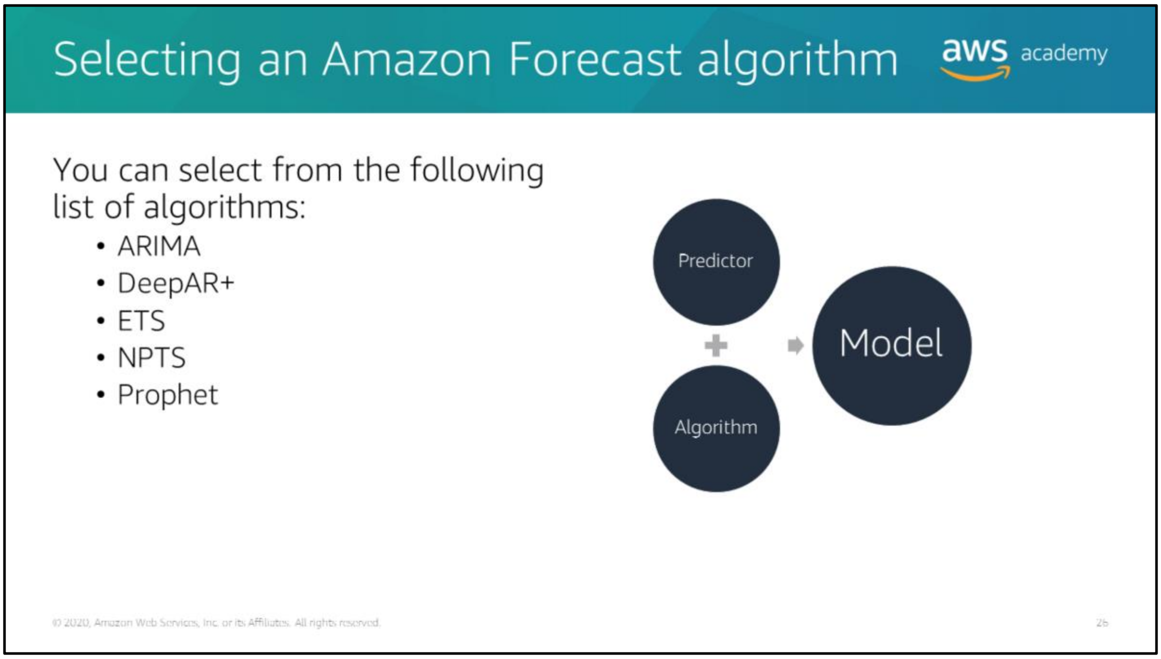
- Webpage ID
- Page views per month
- Timestamp

Related data that creates a more useful forecast includes:

- Page category (such as navigation, or content category)
- Geographic identifier for web client

You might also need the following metadata:

- Region
- Sales promotion information



Selecting an Amazon Forecast algorithm

aws academy

You can select from the following list of algorithms:

- ARIMA
- DeepAR+
- ETS
- NPTS
- Prophet

Predictor

Algorithm

Model

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

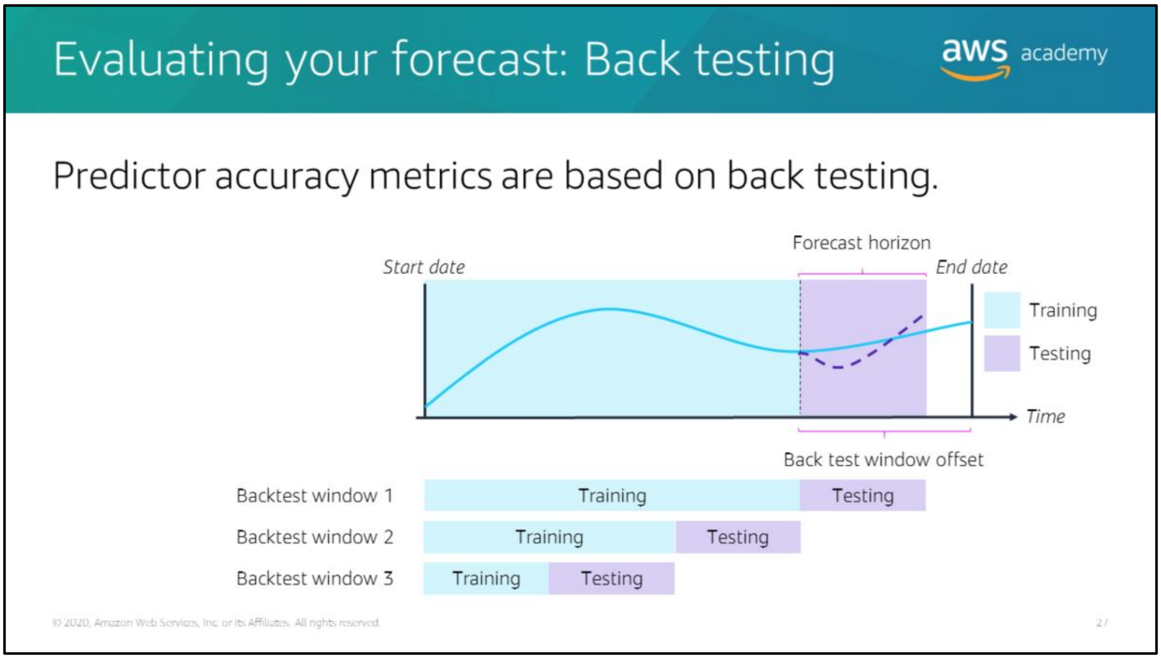
26

Amazon Forecast predictors use an algorithm to train a model. They then use the model to make a forecast by using an input dataset group. To help you get started, Amazon Forecast provides the following predefined algorithms:

- ARIMA
- DeepAR+
- ETS
- NPTS
- Prophet

You can also use the *AutoML* feature, which tries all the algorithms to see which one is the best at predicting data.

For more information about Amazon Forecast algorithms, see the topic Choosing an Amazon Forecast Algorithm in the AWS documentation.



Evaluating your forecast: Back testing

Predictor accuracy metrics are based on back testing.

Backtest window 1: Training | Testing

Backtest window 2: Training | Testing

Backtest window 3: Training | Testing

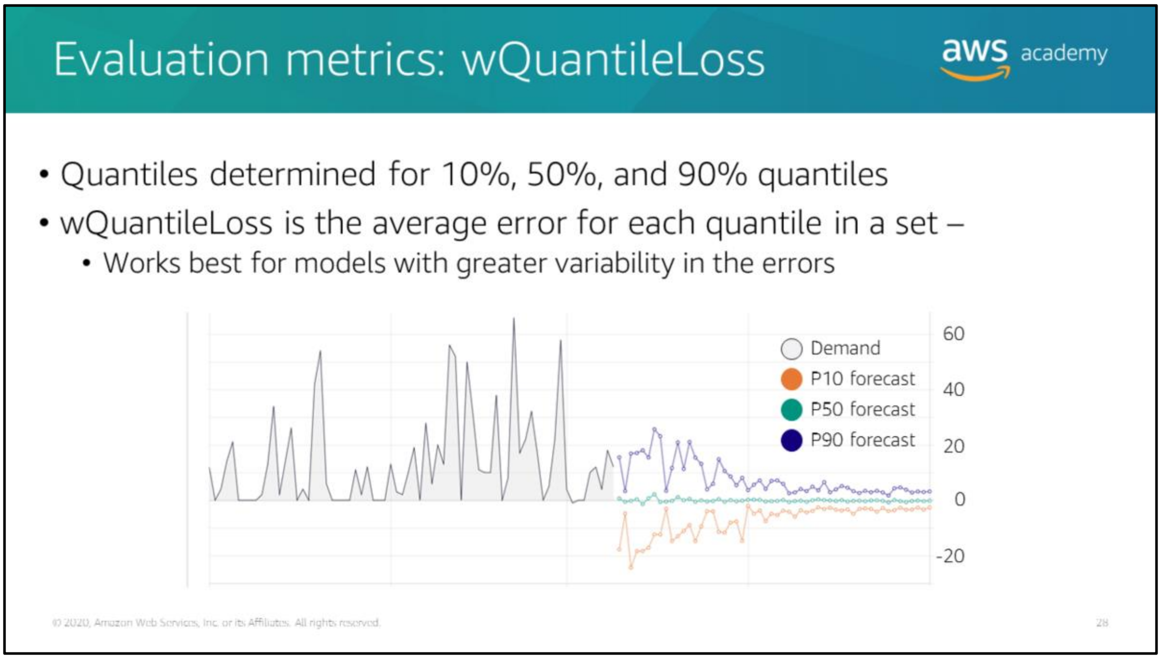
27

When you prepare data for training in machine learning, you typically hold back data to use when you validate and score the model. The data that you hold back is usually a random sample of your available data. With time series data, you must process your data differently because of a correlation between time.

When you import your data, Amazon Forecast breaks it into training and test datasets, which the diagram shows. The training data is used to train the model, which is then tested against the data that was held back. You can specify multiple *back test windows*, which will split the data multiple times, train the model, and use metrics to determine which model gives the best results. The default back test window is *1*.

You can change how Amazon Forecast splits the data by setting the *BackTestWindowOffset* parameter when you create the predictor. If you don't set this value, the algorithms use default values.

After you have trained a model, you will need to measure its accuracy, which you will learn about next.



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

28

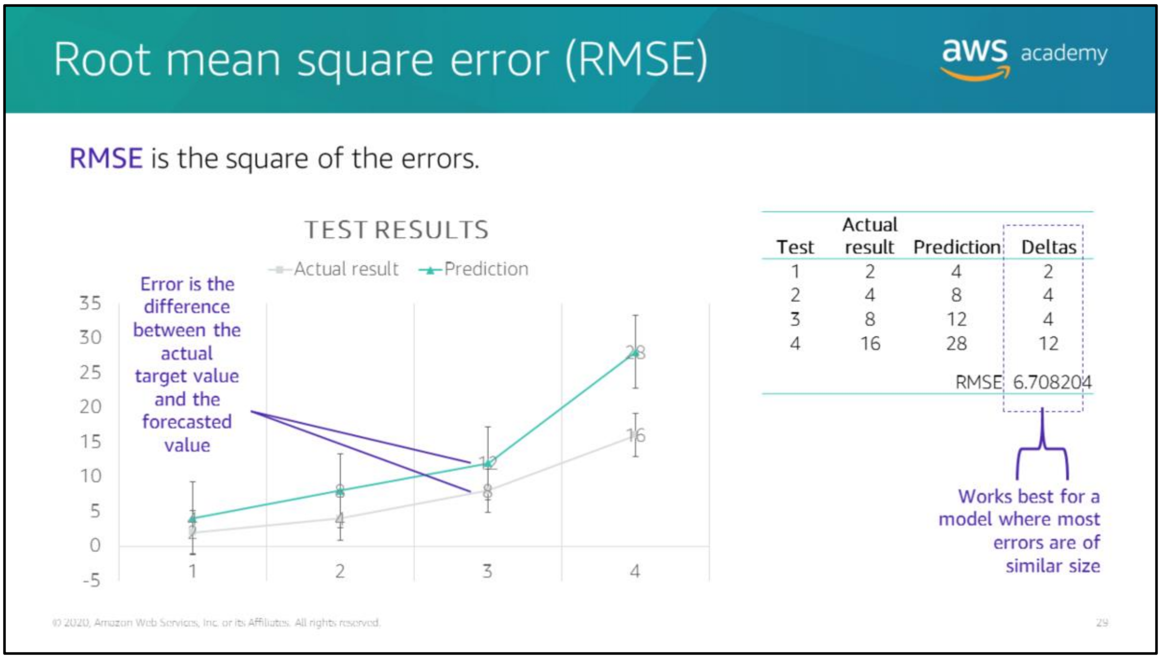
The first Amazon Forecast evaluation metric is the weighted quantile loss (wQuantileLoss). When Amazon Forecast creates a forecast, it provides probabilistic predictions at three distinct quantiles—10 percent, 50 percent, and 90 percent. These prediction quantiles show you how much uncertainty is associated with each forecast.

A P10 quantile predicts that, 10 percent of the time, the true value will be less than the predicted value. For example, suppose that you are a retailer. You want to forecast product demand for winter gloves that sell well only during the fall and winter. Say that you don't have sufficient storage space and the cost of invested capital is high, or that the price of being overstocked on winter gloves concerns you. Then, you might use the P10 quantile to order a relatively low number of winter gloves. You know that the P10 forecast overestimates the demand for your winter gloves only 10 percent of the time, so you will be sold out of your winter gloves for 90 percent of the time.

A P50 quantile predicts that 50 percent of the time, the true value will be less than the predicted value. Continuing the winter gloves example, say you know that there will be a moderate amount of demand for the gloves, and you aren't concerned about being overstocked. Then, you might choose to use the P50 quantile to order gloves.

A P90 quantile predicts that 90 percent of the time, the true value will be less than the predicted value. Suppose you determine that being understocked on gloves will result in large amounts of lost revenue—for example, the cost of not selling gloves is extremely high or the cost of invested capital is low. In this case, you might choose to use the P90 quantile to order gloves.

Amazon Forecast also calculates the associated loss (error) at each quantile. Weighted quantile loss (wQuantileLoss) calculates how far off the forecast a certain quantile is from actual demand in either direction. Lower wQuantileLoss metrics mean that the model's forecasts are more reliable.



Root mean square error (RMSE)

aws academy

RMSE is the square of the errors.

Error is the difference between the actual target value and the forecasted value

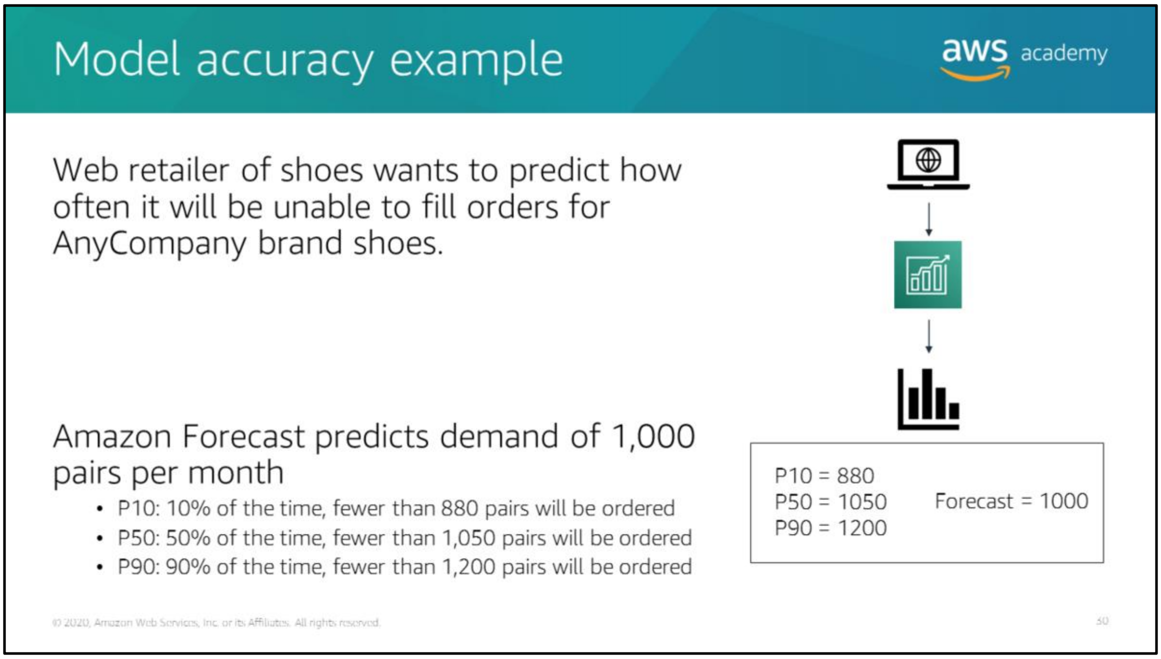
Test	Actual result	Prediction	Deltas
1	2	4	2
2	4	8	4
3	8	12	4
4	16	28	12
		RMSE	6.708204

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

29

The root mean square error (RMSE) is another method for evaluating the reliability of your forecasts. Like wQuantileLoss, RMSE calculates how far off the forecasted values were from the actual test data.

The RMSE finds the difference between the actual target value in the dataset and the forecasted value for that time period, and it then squares the differences. The example shows how to calculate RMSE. The RMSE value represents the standard deviation of the prediction errors. This test is good for forecast validity when the errors are mostly of the same size (that is, there aren't many outliers). Lower RMSE metrics indicate that the model's forecasts are more reliable.



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

30

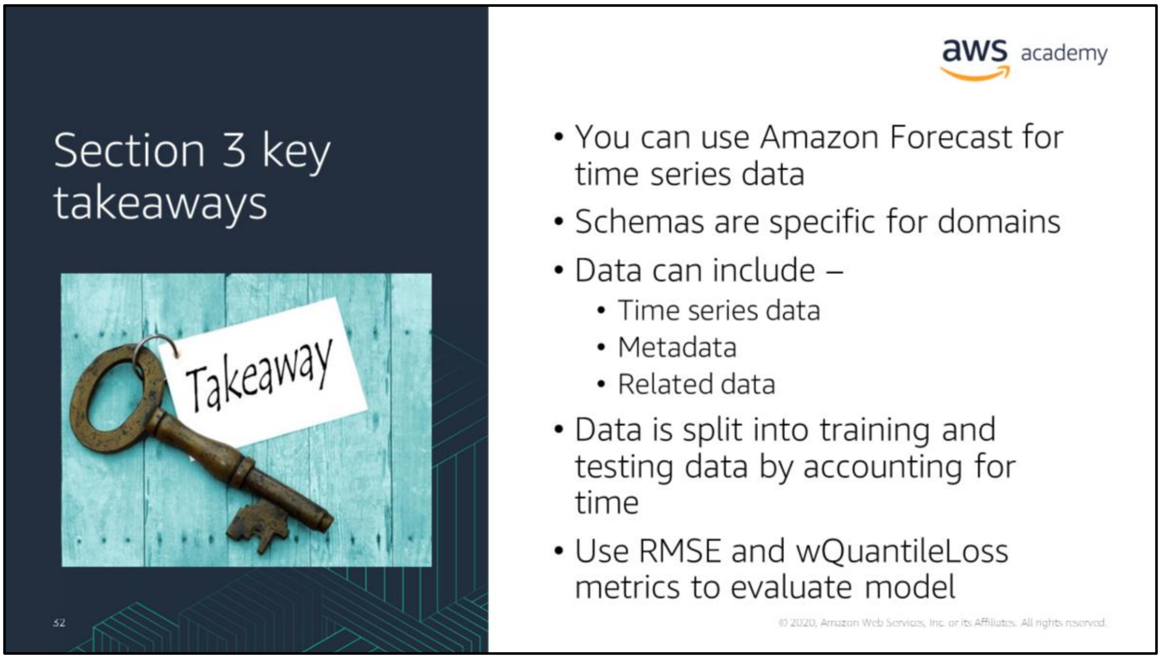
This example shows how a web retailer might use the accuracy metrics to evaluate a forecast. The retailer wants to predict the demand for sales of a particular brand of shoes. They input the sales records for this brand into Amazon Forecast to create a predictor.

The predictor provides a forecasted demand of 1,000 pairs with the P10, P50, and P90 values. The wQuantileLoss values indicate that 10 percent of the time (P10), fewer than 880 pairs will be sold. Next, 50 percent of the time (P50), fewer than 1,050 pairs will be sold. Finally, 90 percent of the time (P90), fewer than 1,200 pairs will be sold. The retailer can then use these values to determine which level of inventory to hold. The determination is based on their assessment of the risk that they can’t fulfill orders, or have excess inventory.

Demonstration: Creating a forecast with Amazon Forecast

51

Your instructor will now either demonstrate how to create a forecast with Amazon Forecast or provide you with access to a recorded demonstration.


aws academy
Takeaway

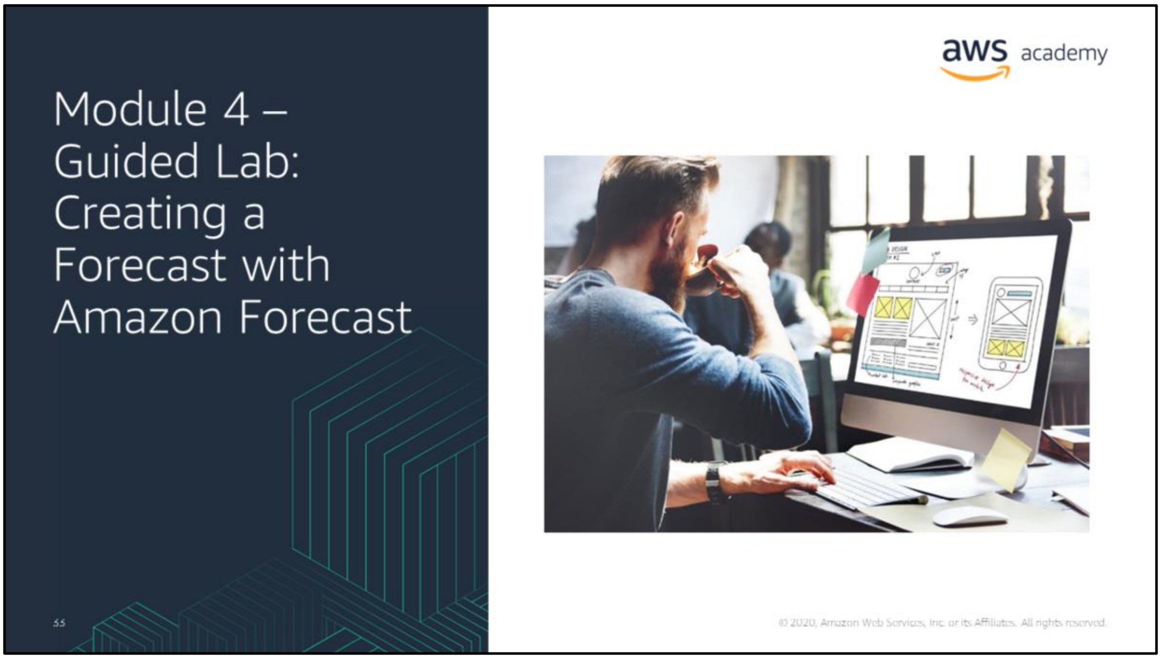
Section 3 key takeaways

- 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

52

Some key takeaways from this section of the module include:

- You can use Amazon Forecast to train and use a model for time series data
- There are specific schemas defined for domains such as retail and EC2 capacity planning, or you can use a custom schema
- You need to supply at least the time series data, but can also provide metadata and related data to add move information to the model
- As with most supervised machine learning problems, your data is split into training and testing data, but this split takes into account the time element
- Use RMSE and wQuantileLoss metrics to evaluate the efficiency of the model



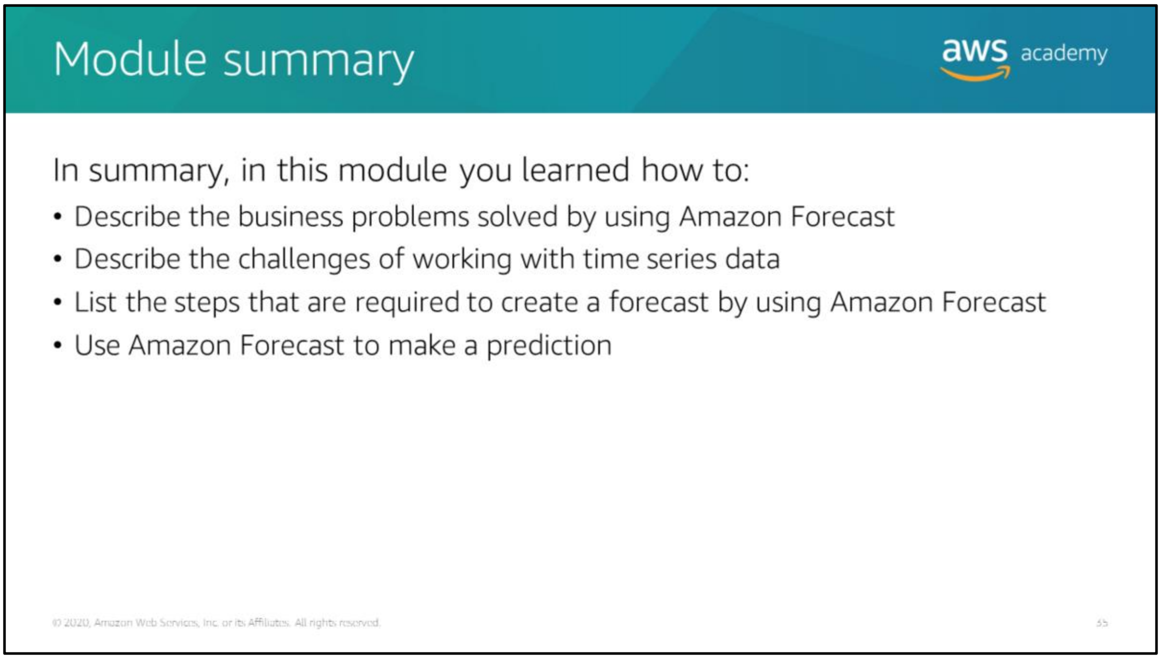
Module 4 – Guided Lab: Creating a Forecast with Amazon Forecast

You will now complete the Module 4 – Guided lab: Creating a Forecast with Amazon Forecast.

Module 4: Introducing Forecasting

Module wrap-up

It's now time to review the module and wrap up with a knowledge check.



Module summary

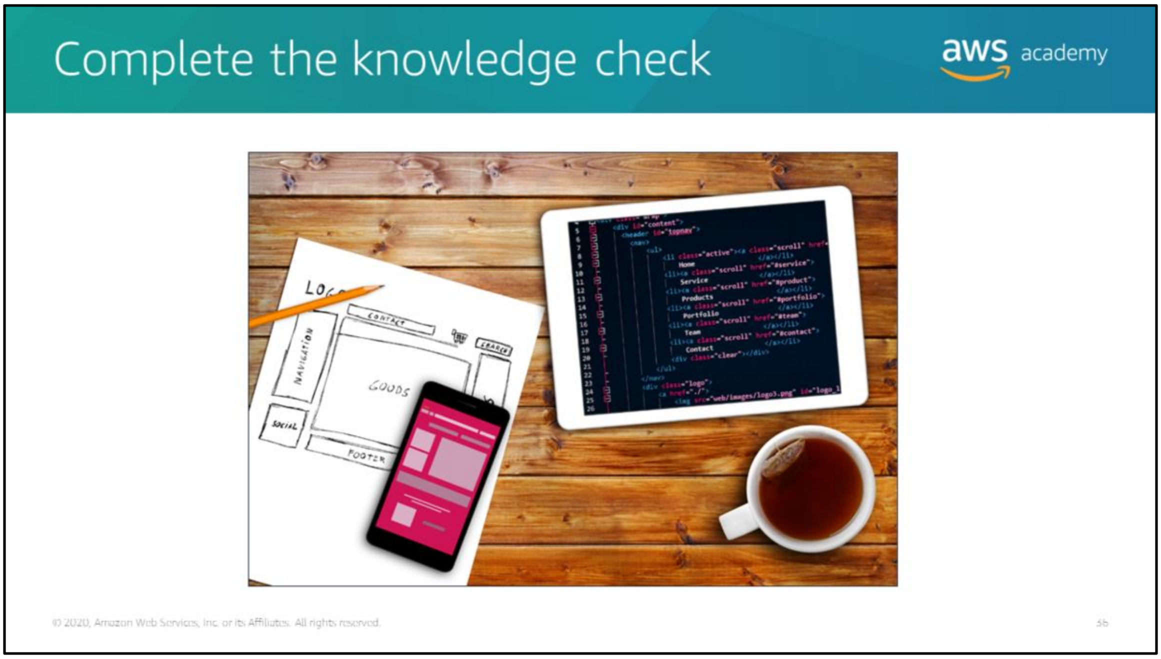
aws academy

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

In summary, in this module you learned how to:

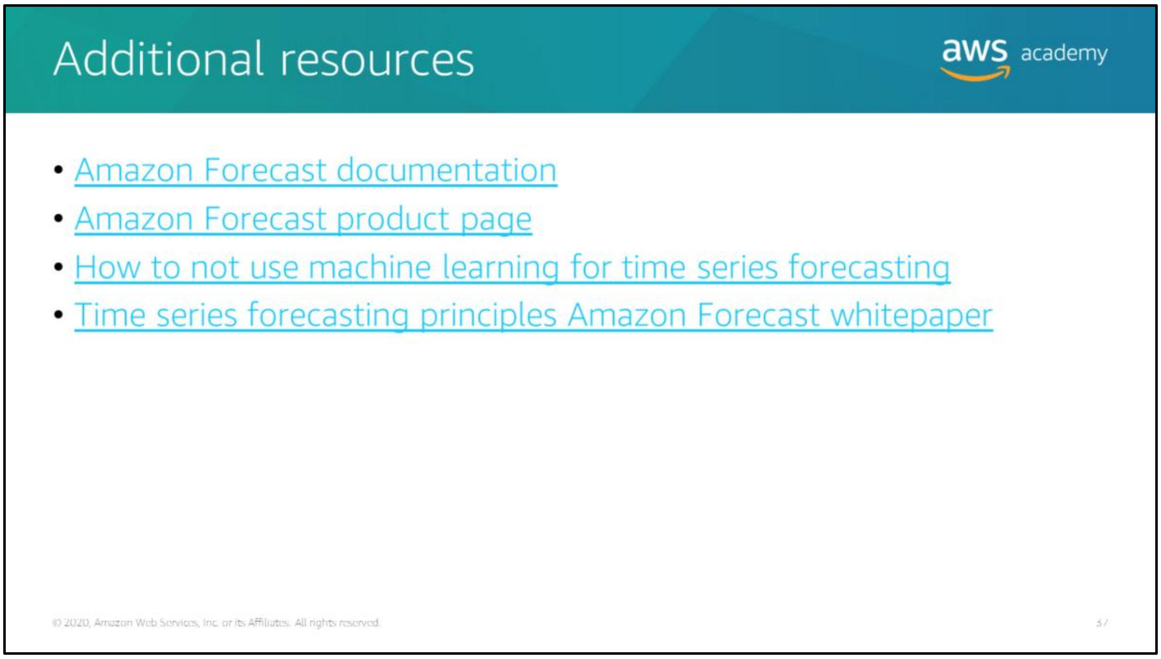
- Describe the business problems solved by using Amazon Forecast
- Describe the challenges of working with time series data
- List the steps that are required to create a forecast by using Amazon Forecast
- Use Amazon Forecast to make a prediction



Complete the knowledge check

56

It is now time to complete the knowledge check for this module.

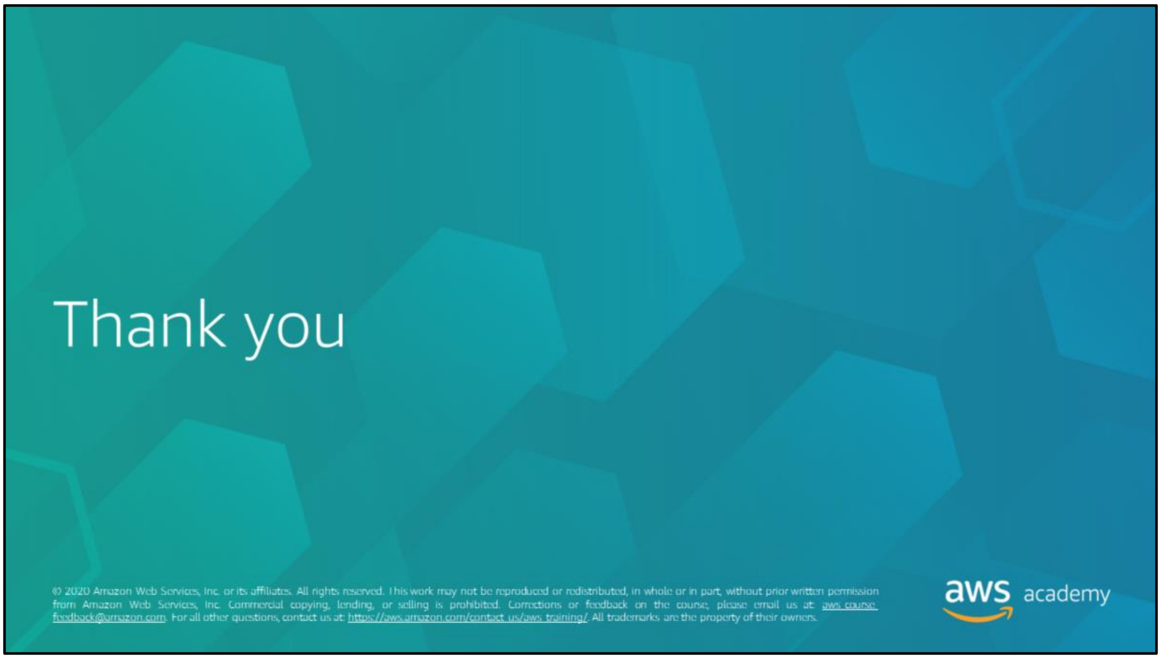


Additional resources

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

If you want to learn more about the topics that are covered in this module, you might find the following additional resources helpful:

- 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

Thanks for participating!