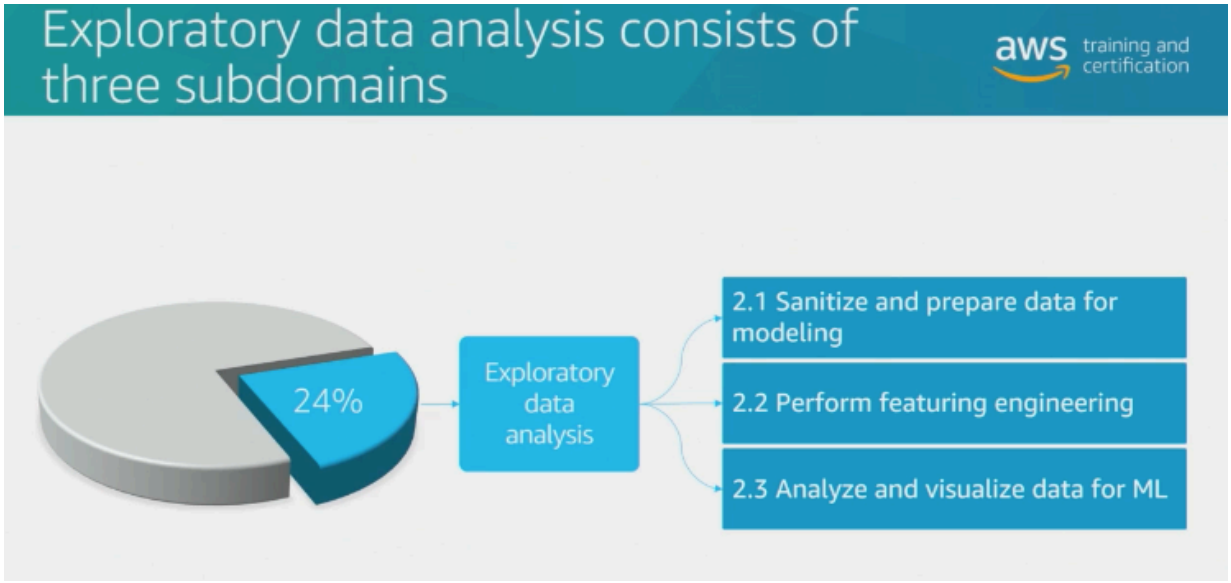
# AWS Exam Readiness - Domain 2: Exploratory Data Analysis

*Module 4 of 9*

# Domain 2: Exploratory Data Analysis




## Domain 2.1: Sanitize and prepare data for modeling

ıge attribution insight by Edwin PM from the Noun Project

You have ingested, transformed, and stored your data in a centralized repository, but it's still messy and not fully understood. You probably have data that is missing, noisy, biased, and/or imbalanced. As a result, you have to sanitize and clean your data so it's more understandable and conducive to effectively training your ML model. This subdomain is about strategies for doing that.

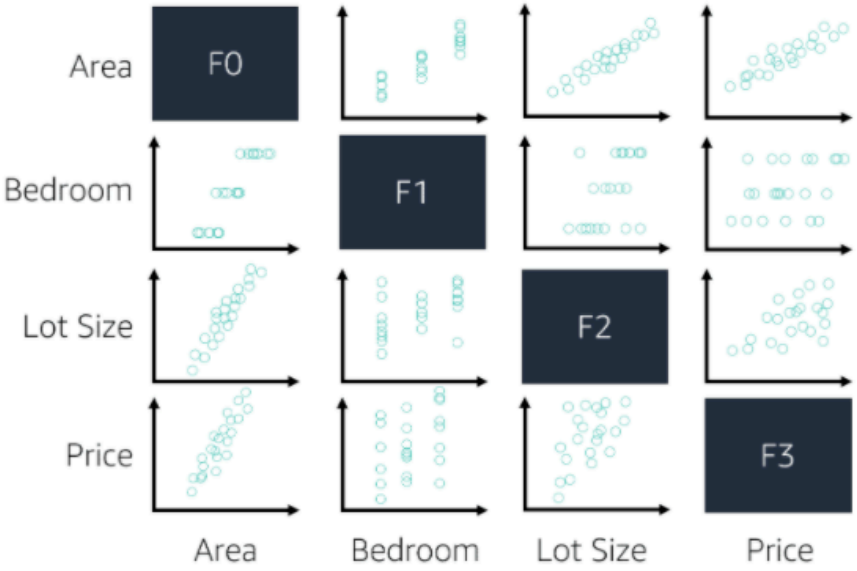
## Use descriptive statistics to better understand your data

The first thing you should do, before cleaning the data, is to use descriptive statistics to better understand your data. Descriptive statistics help you gain valuable insights into your data so that you can more effectively preprocess the data and prepare it for your ML model. Descriptive statistics can be organized into a couple of different categories. Click on the image below to learn more.


# Identifying correlations is important, because they can impact model performance

For cases when you have multiple variables or features, you may want to look at the correlations between them. It's important to identify correlations between attributes, because high correlation between two attributes can sometimes lead to poor model performance. When features are closely correlated and they're all used in the same model to predict the response variable, there could be problems—for example, the model loss not converging to a minimum state. So be aware of highly correlated features in your dataset.

# Scatter plots visualize relationships between numerical variables




When you have more than two numerical variables in a feature dataset and you want to understand their relationship, a scatter plot is a good visualization tool to use. It can help you spot special relationships among those variables. In this plot, for instance, you can see that for some of the variables, there are pretty strong linear relationships, but for other variables, the linear relationship is not as strong.

## Correlation matrices help you quantify the linear relationships among variables

| | F0 | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 |
|---|---|---|---|---|---|---|---|---|---|---|
| F0 | 1.00 | 0.32 | 1.00 | 0.99 | 0.17 | 0.56 | -0.80 | 0.85 | 0.22 | -0.23 |
| F1 | 0.59 | 1.00 | 0.33 | 0.32 | -0.02 | 0.24 | 0.30 | 0.29 | 0.07 | -0.08 |
| F2 | 1.00 | 0.33 | 1.00 | 0.93 | 0.21 | 0.56 | 0.72 | 0.89 | 0.18 | -0.26 |
| F3 | 0.99 | 0.32 | 0.93 | 1.00 | 0.18 | 0.50 | 0.69 | 0.82 | 0.15 | -0.28 |
| F4 | 0.17 | -0.02 | 0.21 | 0.18 | 1.00 | 0.66 | 0.52 | 0.55 | 0.56 | 0.58 |
| F5 | 0.56 | 0.24 | 0.56 | 0.50 | 0.66 | 1.00 | -0.90 | 0.83 | 0.60 | 0.57 |
| F6 | -0.80 | 0.30 | 0.72 | 0.69 | 0.52 | -0.90 | 1.00 | 0.92 | 0.50 | 0.34 |
| F7 | 0.85 | 0.29 | 0.89 | 0.82 | 0.55 | 0.83 | 0.92 | 1.00 | 0.46 | 0.17 |
| F8 | 0.22 | 0.07 | 0.18 | 0.15 | 0.56 | 0.60 | 0.58 | 0.46 | 1.00 | 0.48 |
| F9 | -0.23 | -0.08 | -0.26 | -0.28 | 0.58 | 0.57 | 0.50 | 0.17 | 0.48 | 1.00 |

When this happens, the question becomes: How can you quantify the linear relationship among these variables? A correlation matrix is a good tool in this situation, because it conveys both the strong and weak linear relationships among numerical variables.

For correlation, it can go as high as one, or as low as minus one. When the correlation is one, this means those two numerical features are perfectly correlated with each other. It's like saying Y is proportional to X. When those two variables' correlation is minus one, it's like saying that Y is proportional to minus X. Any linear relationship in between can be quantified by the correlation. So if the correlation is zero, this means there's no linear relationship—but it does not mean that there's no relationship. It's just an indication that there is no linear relationship between those two variables.

### Sanitize your data

Now that you understand your data, it's time to take a closer look at it and begin cleaning it up. Here are a few ways of doing that.

### Standardize language and grammar

Data can be messy in several ways. For instance, maybe your algorithm expects to see data written in English, but there are some words in your dataset from different languages. Or maybe there are special characters in some of the words, or even just a lot of space between words. The key is to make sure you are standardizing your data. If your algorithm requires English, make sure it's all in English.

The same goes for grammatical structure. For example, convert your text data into all lowercase so the same word isn't treated as two different words just because of its capitalization.

| ID | Survey Response |
|---|---|
| 1 | This is grrreat! |
| 2 | This is grrreat! |
| 3 | Y E s |
| 4 | ur $0 L33t! |
| 5 | ¿Qué? |
| 6 | 或者 |
| 7 | احب ذلك |

### Make sure the data is on the same scale

Your dataset might also include data that is on very different scales. For example, here we have one column called Length, but that column has different units for data, like kilometers, meters, and miles. This is a common occurrence in many numerical datasets, especially if your dataset is a result of merging data from multiple sources.

| ID | Survey Response | ID | Length |
|---|---|---|---|
| 1 | This is grrreat! | 1 | 40 km |
| 2 | This is grrreat! | 2 | 100 m |
| 3 | Y E s | 3 | 100 mi |
| 4 | ur $0 L33t! | 4 | 74 m |
| 5 | ¿Qué? | 5 | 74 ft |
| 6 | 或者 | 6 | 29 in |
| 7 | احب ذلك | 7 | 1092 nm |

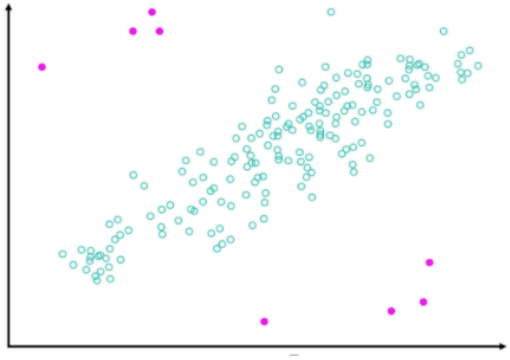
## Make sure a column doesn't include multiple features

An even messier example is when you have a column of data that has multiple features represented. For instance, a column called Measurement includes temperatures, distance, time, and other numerical values. In this situation, you have to reshape the data so that each specific feature will be in its own column.

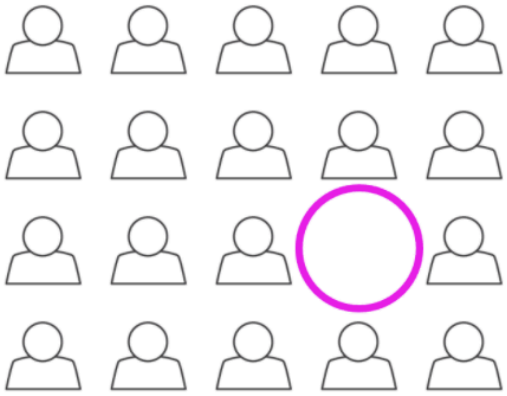
| ID | Survey Response | ID | Length | ID | Measurement |
|---|---|---|---|---|---|
| 1 | This is grrreat! | 1 | 40 km | 1 | 96 km |
| 2 | This is grrreat! | 2 | 100 m | 2 | twenty |
| 3 | Y E s | 3 | 100 mi | 3 | 5:40:27 |
| 4 | ur $0 L33t! | 4 | 74 m | 4 | 735 $cm^3$ |
| 5 | ¿Qué? | 5 | 74 ft | 5 | 2 cats |
| 6 | 或者 | 6 | 29 in | 6 | 44 °C |
| 7 | احب ذلك | 7 | 1092 nm | 7 | 346 Mb/s |

**You might also need to clean your data based on any outliers that may exist**

Outliers are points in your dataset that lie at an abnormal distance from other values. They are not always something you want to clean up, because they can add richness to your dataset. But they can also make it harder to make accurate predictions, because they skew values away from the other, more normal, values related to that feature. Moreover, an outlier can also indicate that the data point actually belongs to another column.



The pink data points in this scatter plot represent outliers



**Missing data also needs to be handled at this stage**

You may also find that you have missing data. For example, some columns in your dataset might be missing data due to a data collection error, or perhaps data was not collected on a particular feature until well into the data collection process. Missing data can make it difficult to accurately interpret the relationship between the related feature and the target variable, so, regardless of how the data ended up being missed, it is important to deal with the issue.

Here are a few ways to fill missing data:

- Remove the columns or rows that include the missing data
- Fill the missing value with the column mean, a zero, or another numerical value using imputation

Neither of these approaches is without trade-offs, so choose carefully based on the data. Generally speaking, if the data is small and you cannot afford to lose too many data points, you should choose one of the imputation techniques.

## Topics related to this subdomain

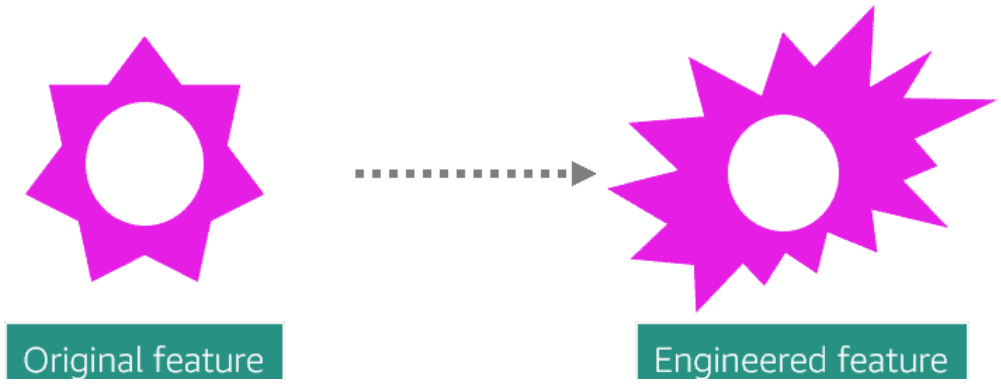
Here are some topics you may want to study for more in-depth information related to this subdomain:

- [ ] Dataset generation
  - Amazon SageMaker Ground Truth
  - Amazon Mechanical Turk
  - Amazon Kinesis Data Analytics
  - Amazon Kinesis Video Streams
- [ ] Data augmentation
- [ ] Descriptive statistics
- [ ] Informative statistics
- [ ] Handling missing values and outliers

## Domain 2.2: Perform feature engineering

### Feature engineering gives your model stronger prediction power

The work doesn't end once you clean your data. The features you have are still raw. In order to make the kinds of predictions you want to make, you need to create new features from the original ones. The goal is to give your model stronger prediction power with these new features. This process is referred to as feature engineering. This subdomain is about understanding the different approaches to performing feature engineering. This section will highlight a small set of feature engineering methods that are commonly used in data science and ML.

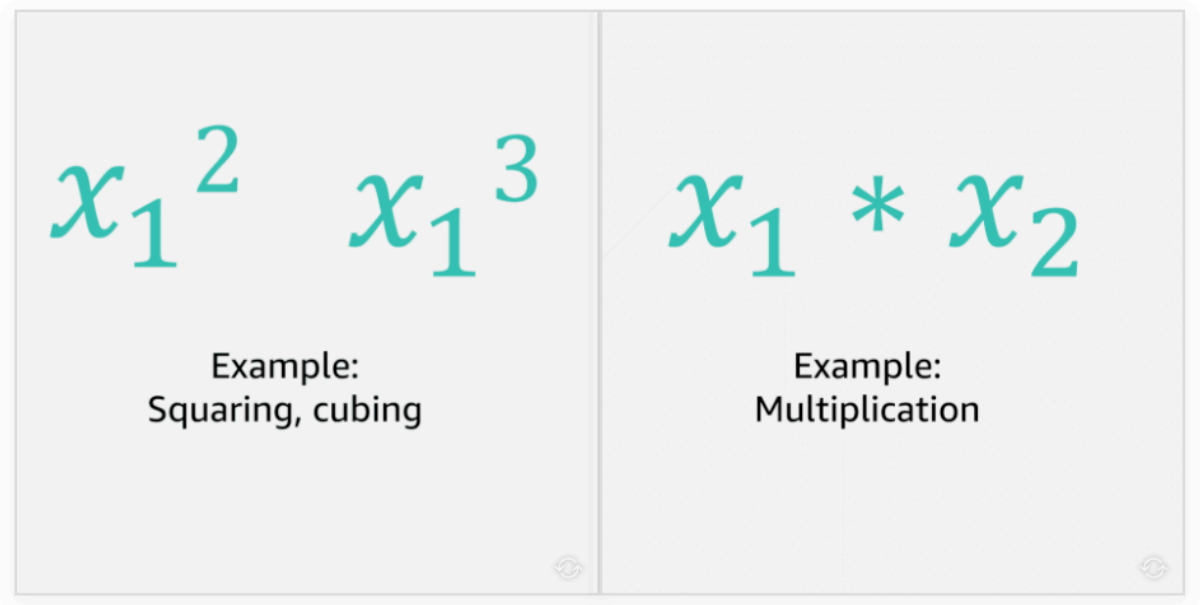



### You may need to perform feature engineering because of the dimensionality of your dataset—particularly if there are too many features for your model to handle

To reduce the number of features, you need to deploy dimensionality reduction techniques like principal component analysis (PCA) or t-distributed stochastic neighbor embedding.

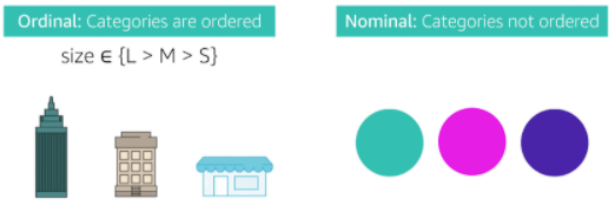
## For numerical features, you can do what is referred to as transformation

One of the examples below is of a multinominal or polynomial transformation, where you take the square and cube of the original feature and use all three columns as separate attributes while training your model. The other is an example of multiplication as a means of transformation.




## You may need to perform feature engineering because of the format of your data

You will often handle categorical data that needs to be converted into numerical data before it can be read by your ML algorithm. Your approach will differ depending on whether your data is ordinal (the categories are ordered) or nominal (categories are not ordered).



## Example of feature engineering

1

| Type | Bedrooms | Area (sq. ft) | Garden Size | Price | Loan Approved |
|---|---|---|---|---|---|
| House | 3 | 2572 | Small | 1372000 | Yes |
| Apartment | 2 | 1386 | N/A | 699000 | No |
| Condo | 3 | 1932 | Large | 800000 | No |
| Condo | 1 | 851 | Medium | 451000 | Yes |
| Apartment | 1 | 600 | N/A | 325000 | No |

Here is a piece of a home mortgage dataset used to predict loan approvals. The dataset includes several features: home type, number of bedrooms, area of home, garden size, and home price. There's also a yes/no value as the target variable, indicating whether a home loan was approved or not.

| Type | Bedrooms | Area (sq. ft) | Garden Size | Price | Loan Approved |
|---|---|---|---|---|---|
| House | 3 | 2572 | Small | 1372000 | Yes = 1 |
| Apartment | 2 | 1386 | N/A | 699000 | No = 0 |
| Condo | 3 | 1932 | Large | 800000 | No = 0 |
| Condo | 1 | 851 | Medium | 451000 | Yes = 1 |
| Apartment | 1 | 600 | N/A | 325000 | No = 0 |

A binary categorical variable, such as the yes/no target variable, can be easily encoded into 1s and 0s. 1s represent "yes" (the loan is approved), and 0s represent "no" (the loan has not been approved).

| Type | Bedrooms | Area (sq. ft) | Garden Size | Price | Loan Approved |
|---|---|---|---|---|---|
| House | 3 | 2572 | Small | 1372000 | 1 |
| Apartment | 2 | 1386 | N/A | 699000 | 0 |
| Condo | 3 | 1932 | Large | 800000 | 0 |
| Condo | 1 | 851 | Medium | 451000 | 1 |
| Apartment | 1 | 600 | N/A | 325000 | 0 |

Some of the features, like number of bedrooms, area of home, and home price, are numerical variables. We'll look at some methods for featuring engineering numerical variables in just a bit.

| Type | Bedrooms | Area (sq. ft) | Garden Size | Price | Loan Approved |
|---|---|---|---|---|---|
| House | 3 | 2572 | Small = 5 | 1372000 | 1 |
| Apartment | 2 | 1386 | N/A = 0 | 699000 | 0 |
| Condo | 3 | 1932 | Large = 20 | 800000 | 0 |
| Condo | 1 | 851 | Med. =10 | 451000 | 1 |
| Apartment | 1 | 600 | N/A = 0 | 325000 | 0 |

Home type and garden size represent categorical features. Garden size, more specifically, represents ordinal data, because Small, Medium, and Large can be represented in an order (note: N/A represents "no garden").

For ordinal variables like this one, you can use a map function in Pandas to convert the text into numerical values. For example, you can define the relative difference for those different categories in the ordinal variable.

Often, the numerical value you provide in the mapping is derived from your business insight of the dataset and the business itself. For the garden size S, you can use 5; for M, use 10; for L, use 20; and for N, use 0.

You can easily apply the map function from Pandas to replace the categorical variable with the numerical value. It is a one-to-one mapping.

| Type | Bedrooms | Area (sq. ft) | Garden Size | Price | Loan Approved |
|---|---|---|---|---|---|
| House | 3 | 2572 | 5 | 1372000 | 1 |
| Apartment | 2 | 1386 | 0 | 699000 | 0 |
| Condo | 3 | 1932 | 20 | 800000 | 0 |
| Condo | 1 | 851 | 10 | 451000 | 1 |
| Apartment | 1 | 600 | 0 | 325000 | 0 |

By contrast, AWS does not recommend encoding nominal variables like home type to numerical data. If you encode this variable or feature into integers, it becomes one, two, and three.

One, two, and three really implies that something has a numerical value. They have order difference, and there is also a magnitude to the difference between the numbers. These additional features are artifacts that do not belong to the original data. And these artifacts may give you the wrong or unexpected results.


| Type | Bedrooms | Area (sq. ft) | Garden Size | Price | Loan Approved |
|---|---|---|---|---|---|
| House | 3 | 2572 | 5 | 1372000 | 1 |
| Apartment | 2 | 1386 | 0 | 699000 | 0 |
| Condo | 3 | 1932 | 20 | 800000 | 0 |
| Condo | 1 | 851 | 10 | 451000 | 1 |
| Apartment | 1 | 600 | 0 | 325000 | 0 |


So how do you encode nominal variables? The one-hot encoding method is a good choice. Here's how it works.

| | Type_House | Type_Apartm. | Type_Condo |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 |
| 3 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 |

In this example, you have the one column called home Type, and three different levels: House, Apartment, and Condo. The data frame has five observations for that particular feature.

With one-hot encoding, you convert this one column of home Type into three columns: a column for House, a column for Apartment, and a column for Condo. You encode each observation with either a 1 or 0: 1 to indicate the home type of that particular observation, or 0 for the other options.

## Numerical data can be scaled to ensure proportionate influence on the prediction

| Type | Bedrooms | Area (sq. ft) | Garden Size | Price | Loan Approved |
|---|---|---|---|---|---|
| House | 3 | 2572 | 5 | 1372000 | 1 |
| Apartment | 2 | 1386 | 0 | 699000 | 0 |
| Condo | 3 | 1932 | 20 | 800000 | 0 |
| Condo | 1 | 851 | 10 | 451000 | 1 |
| Apartment | 1 | 600 | 0 | 325000 | 0 |

Let's go back to numerical variables for a minute. With numerical variables, you may need to scale your data to avoid having one feature with more importance than the others due to their original range. In this last example, for the typical numerical features like home price, number of bedrooms, and area of home, you need scaling.

For each column in this particular dataset, you want the value to be between minus one and plus one. The main reason for scaling is that the range of these features is dramatically different.

For example, here, Bedrooms has a value of one, two, or three. There are not many bedrooms in the house, but on the other hand, the price of the house ranges from around $300,000 to over $1.25 million. That's a huge difference in scale, and that can give disproportionate influence to larger scaled variables.

## Common techniques for scaling

So how do we do it, exactly? How can we align different features into the same scale?

Keep in mind that not all ML algorithms will be sensitive to different scales of inputted features. Here is a collection of commonly used scaling and normalizing transformations that we usually use for data science and ML projects:

- Mean/variance standardization
- MinMax scaling
- Maxabs scaling
- Robust scaling
- Normalizer

### Topics related to this subdomain

Here are some topics you may want to study for more in-depth information related to this subdomain:

- [ ] Scaling
- [ ] Normalizing
- [ ] Dimensionality reduction
- [ ] Date formatting
- [ ] One-hot encoding

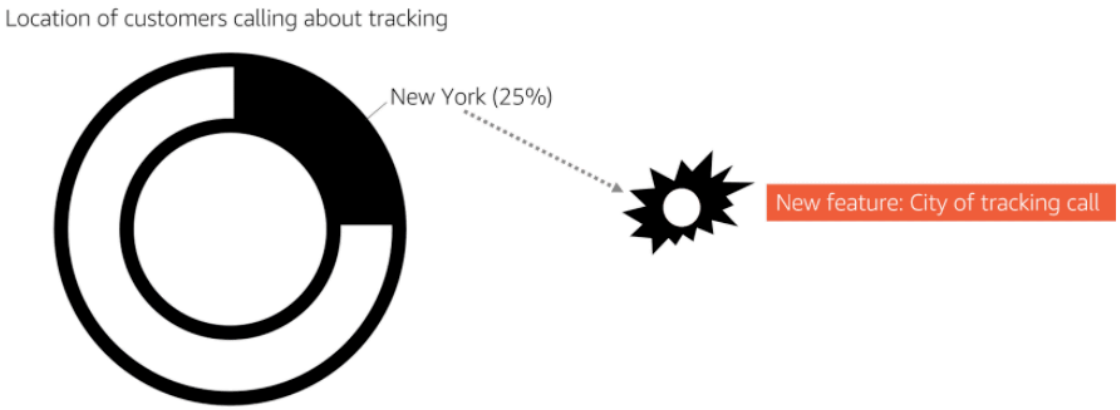
---

### Domain 2.3: Analyze and visualize data for ML

## Visualization helps you better understand your features and their relationships

Part of the exploratory data analysis phase of the ML pipeline is analyzing and visualizing your data, which helps you better understand your features and the relationships among features. Visualization techniques are key tools in your toolbox when sanitizing your data and performing feature engineering.

Visualization techniques include visualizing averages and summary statistics using line charts, histograms, and an ever-expanding catalog of custom visualizations. This subdomain focuses on assessing your understanding of these and other techniques, and tests your ability to analyze the visualized data to make informed decisions from it.

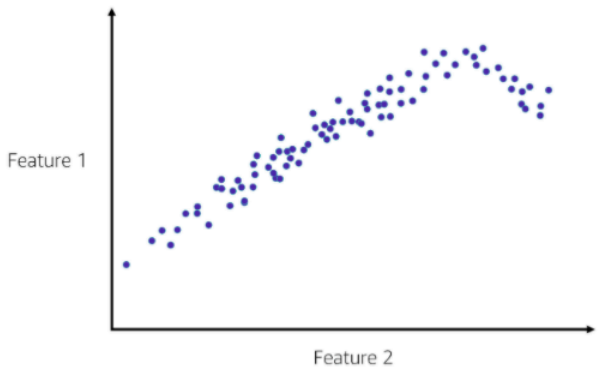



## Visualizations help give you a better idea of what's inside a particular feature and help you answer questions like these:

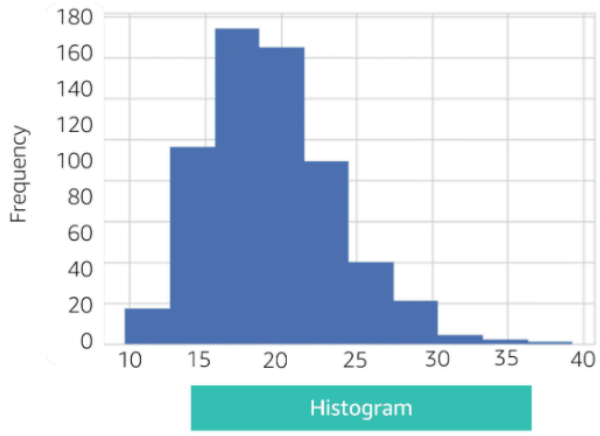
- What's the range of the data?
- What's the peak of the data?
- Are there any outliers?
- Are there any interesting patterns in the data?

Data visualization will also help you determine whether you need to clean and preprocess your data before model training.

## Common visualization techniques include scatter plots and histograms




A scatter plot is used to visualize the relationship between two variables. In this case, we have Feature 1 and Feature 2: two numerical variables. As you can see, there are plots scattered around. Even though the correlation among them may not be that high because the data is scattered around quite a bit, there may be some relatively positive relationships between the two variables.




You also can look at the histogram, which shows the distribution of an individual feature in your dataset.

By looking at the histogram of an individual feature, you can see the overall behavior of that particular variable. Is it normally distributed, is there only that one peak, are there multiple peaks in the data? Or you can spot other high-level important features, like skewness for that particular variable.

Additionally, for the histogram, you're seeing all the observations for that particular variable. Because you are extracting the information from all the data for that specific variable, you get the entire overview for that piece of information.

## Topics related to this subdomain

Here are some topics you may want to study for more in-depth information related to this subdomain:

- [ ] Scatter plots
- [ ] Box plots
- [ ] Histograms
- [ ] Scatter matrix
- [ ] Correlation matrix
- [ ] Heatmaps
- [ ] Confusion matrix

## Walk-through of sample questions

In this section, you'll have a chance to answer and walk through the solutions to two different sample questions. The questions reflect some of the design and technical content you may see on the exam. The videos below will give you the answers to each question by walking you through the test-taking strategies presented to you earlier in this course.

## Question 1

Answer the question below before watching the corresponding solution video.

A team of data scientists in a company focusing on security and smart home devices created an ML model that can classify guest types at a front door using a video doorbell. The team is getting an accuracy of 96.23% on the validation dataset.

However, when the team tested this model in production, images were classified with a much lower accuracy. That was due to weather: The changing seasons had an impact on the quality of production images.

What can the team do to improve their model?

- (•) Use data augmentation techniques to add more images so that the model can generalize better.
- ( ) Normalize the dataset so that the features are on the same scale to help with convergence.
- ( ) Give the model more time to train so it learns the key features more effectively.
- ( ) Use a different convolutional neural network (CNN) algorithm that will give a better model accuracy.

Correct

## Question 2

Answer the question below before watching the corresponding solution video.

A team of data scientists in a financial company wants to predict the risk for their incoming customer loan applications. The team has decided to do this by applying the XGBoost algorithm, which will predict the probability that a customer will default on a loan. In order to create this solution, the team wants to first merge the customer application data with demographic and location data before feeding it into the model.

However, the dimension of this data is really large, and the team wants to keep only those features that are the most relevant to the prediction.

What techniques can the team use to reach the goal? (Select TWO.)

- [ ] Use dropout to remove some of the columns during training
- [ ] Use t-Distributed Stochastic Neighbor Embedding (t-SNE)
- [x] Use clustering to find the salient clusters and use them as features
- [x] Use the principal component analysis (PCA) algorithm
- [ ] Use AWS Glue feature_tokenizer function to automatically drop the irrelevant features

- ✗ Use dropout to remove some of the columns during training
- ✓ Use t-Distributed Stochastic Neighbor Embedding (t-SNE)
- ✗ Use clustering to find the salient clusters and use them as features
- ✓ Use the principal component analysis (PCA) algorithm
- ✗ Use AWS Glue feature_tokenizer function to automatically drop the irrelevant features

A team of Data Scientists in a financial company wants to predict the risk for its incoming customer loan applications. The team has decided to do this by applying the XGBoost algorithm, which will predict the probability that a customer will default on a loan. In order to create this solution, the team wants to first merge the customer application data with demographic and location data before feeding it into the model.

However, the dimension of this data is really large, and the team wants to keep only those features that are the most relevant to the prediction.

**Domain quiz**

Take this short quiz to test your understanding of some of the topics related to this domain and experience the types of questions that will be on the exam. We recommend you use the test-taking strategies presented earlier in this course when completing the questions. Click the link below to begin.

aws training and certification

A Machine Learning Engineer is creating and preparing data for a linear regression model. However, while preparing the data, the Engineer notices that about 20% of the numerical data contains missing values in the same two columns. The shape of the data is 500 rows by 4 columns, including the target column.

How could the Engineer handle the missing values in the data? (Select TWO.)

- [ ] Fill the missing values with zeros
- [x] Add regularization to the model
- [ ] Remove the columns containing the missing values
- [x] Impute the missing values using regression
- [ ] Remove the rows containing the missing values

Fill the missing values with zeros

✗ **Add regularization to the model**

Remove the columns containing the missing values

✓ **Impute the missing values using regression**

Remove the rows containing the missing values

A social networking organization wants to analyze all the comments and likes from its users to flag offensive language on the site. The organization's data science team wants to use a Long Short-term Memory (LSTM) architecture to classify the raw sentences from the comments into one of two categories: offensive and non-offensive.

What should the team do to prepare the data for the LSTM?

- ○ Vectorize the sentences. Transform them into numerical sequences with a padding. Use the sentences as the input.
- ◉ Convert the individual sentences into sequences of words. Use those as the input.
- ○ Convert the individual sentences into numerical sequences starting from the number 1 for each word in a sentence. Use the sentences as the input.
- ○ Vectorize the sentences. Transform them into numerical sequences. Use the sentences as the input.

Vectorize the sentences. Transform them into numerical sequences with a padding. Use the sentences as the input.

✕ **Convert the individual sentences into sequences of words. Use those as the input.**

Convert the individual sentences into numerical sequences starting from the number 1 for each word in a sentence. Use the sentences as the input.

Vectorize the sentences. Transform them into numerical sequences. Use the sentences as the input.

A Data Scientist created a correlation matrix between nine variables and the target variable. The correlation coefficient between two of the numerical variables, variable 1 and variable 5, is -0.95.

How should the Data Scientist interpret the correlation coefficient?

- ○ Variable 1 does not have any influence on variable 5
- ◉ As variable 1 increases, variable 5 decreases
- ○ As variable 1 increases, variable 5 increases
- ○ The data is not sufficient to make a well-informed interpretation

**2/4** **50.0%**