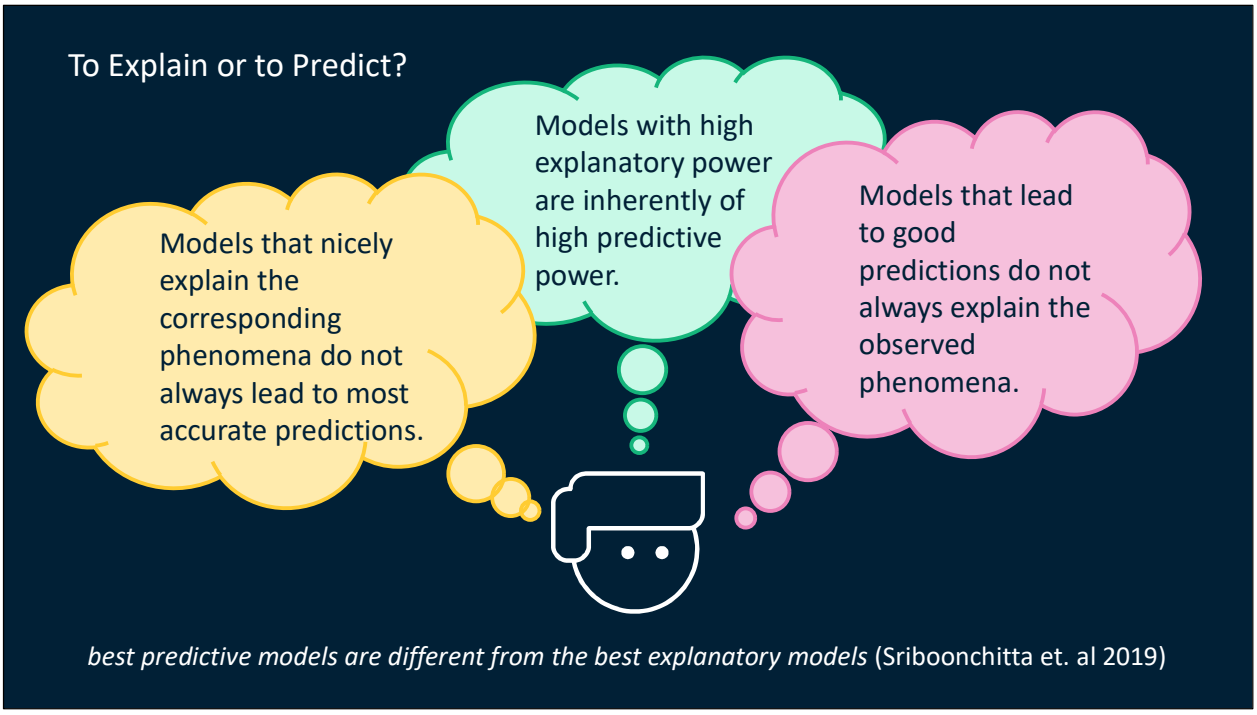
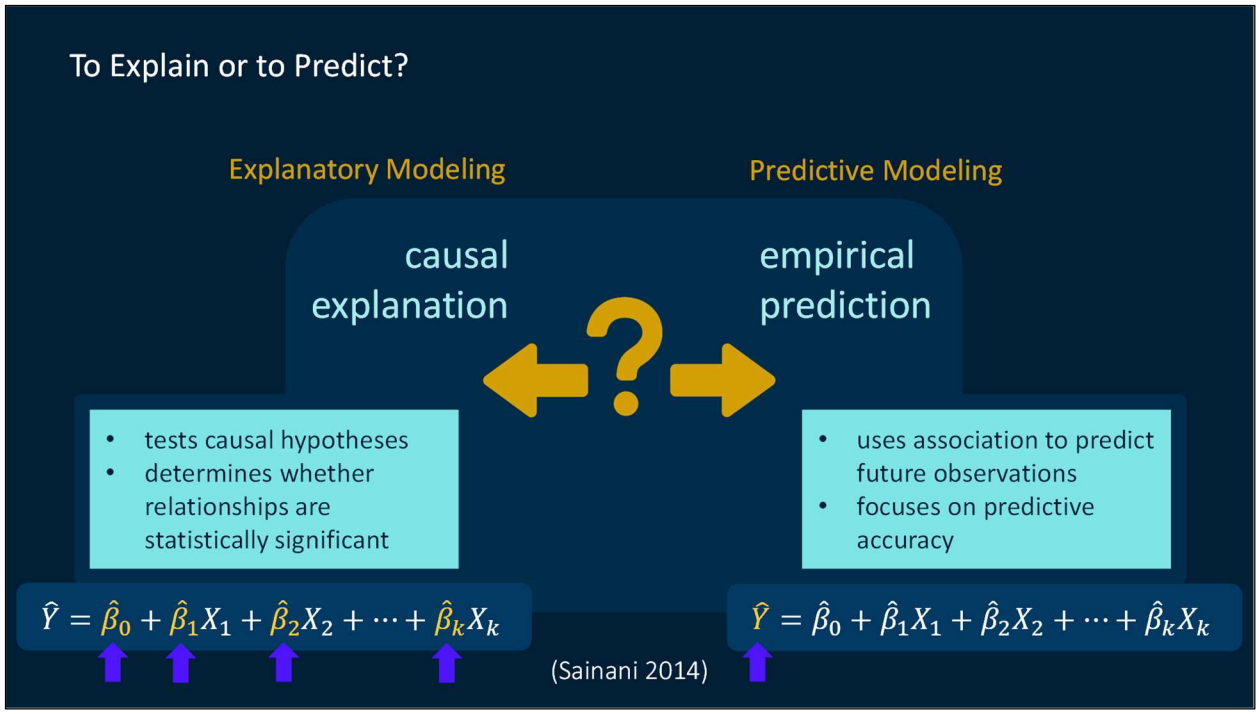
Lesson 4 Predictive Modeling Using Logistic Regression

4.1 Introduction to Predictive Modeling



In statistics, it is implicitly assumed that models that are the best predictors also have the best explanatory power. Of course, an explanatory model can be used for prediction - and likewise, a predictive model can be used for explanation.

However, the best predictive models are often different from the best explanatory models. In practice, models that lead to good predictions do not always explain the observed phenomena. Vice versa, models that nicely explain the corresponding phenomena do not always lead to most accurate predictions.

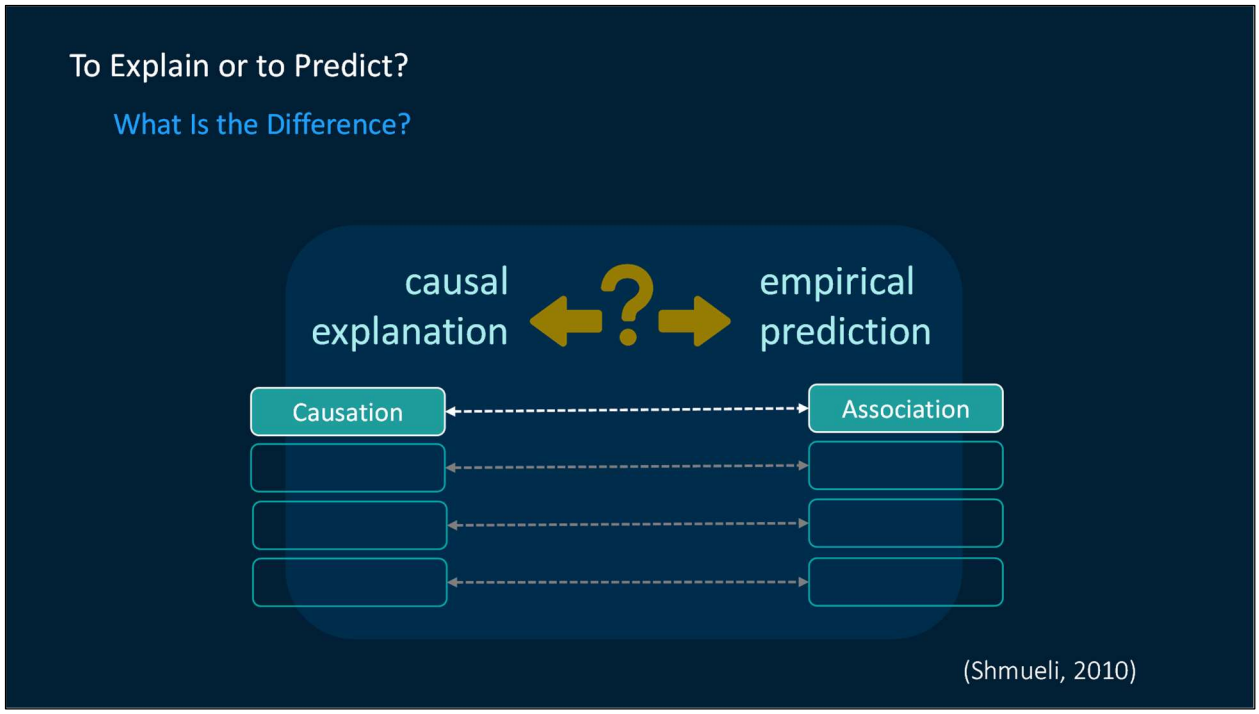


Earlier in this course, you learned to build models for the purpose of understanding and explaining the relationships between a response and a set of predictors. But model building works differently for purely predictive models. This distinction affects every aspect of model building and evaluation.

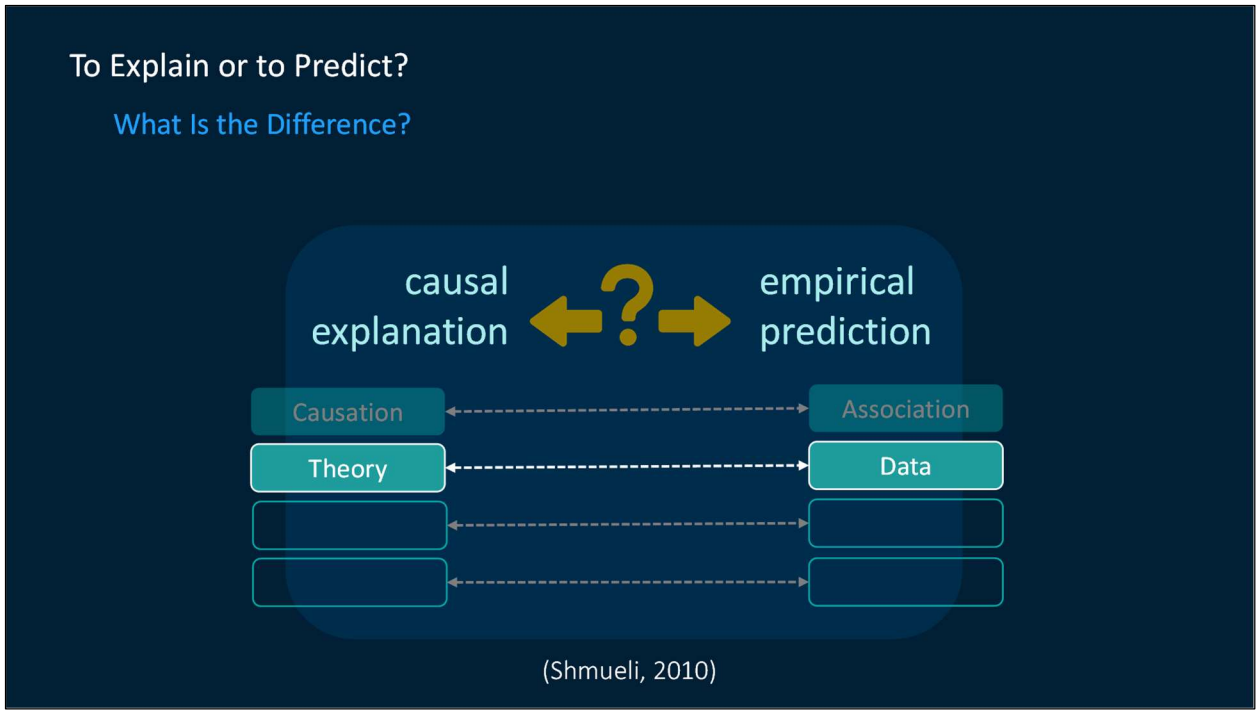
Explanatory modeling and predictive modeling have distinct goals that they are aimed at causal explanation and empirical prediction, respectively.

In explanatory modeling, statistical models are applied to data in order to test causal hypotheses. Thus, looking at the model equations, explanatory modeling focuses on the estimates of the beta coefficients.

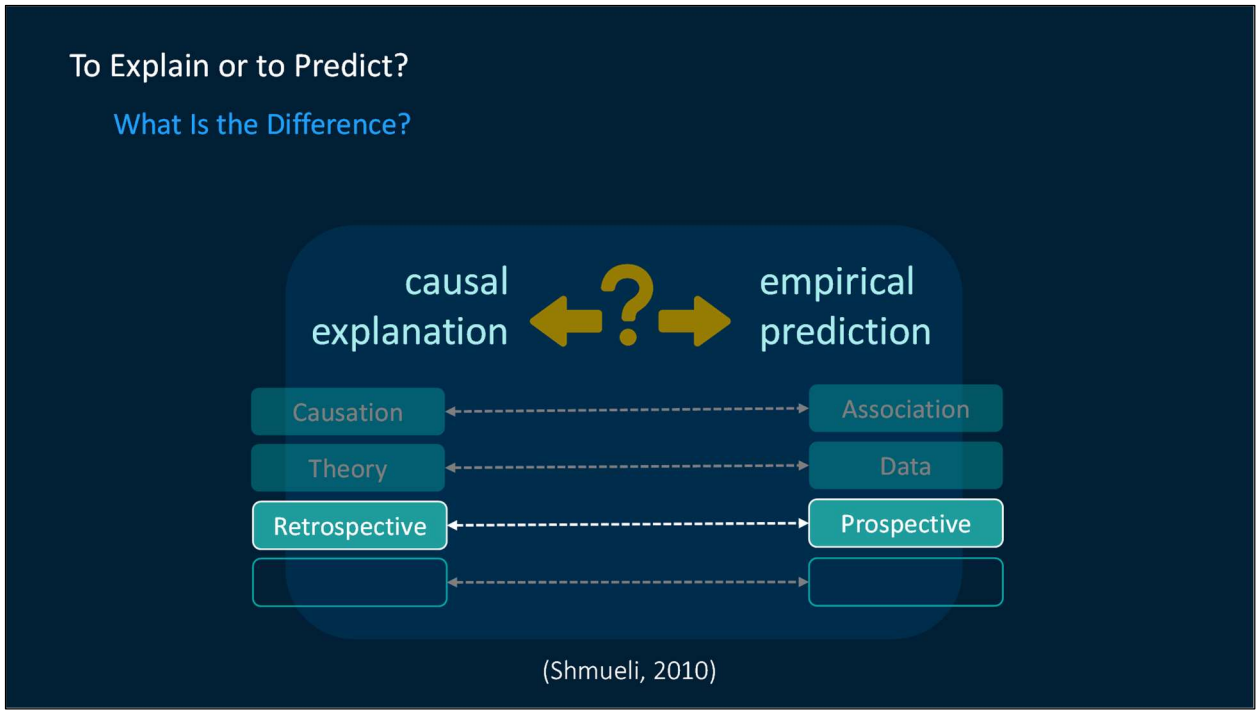
In predictive modeling our interest is different. Predictive modeling is a process of applying a statistical model or data mining and machine learning algorithm to data for the purpose of predicting new or future observations. Here, the goal is to use the associations between predictors and the response to generate good predictions for future outcomes. As a result, predictive models are created very differently from explanatory models. The primary goal is predictive accuracy. Thus, the focus is not so much on the parameters of the model as it is in the predictions of observations. In the model equation, these are represented on the left side of the equation.



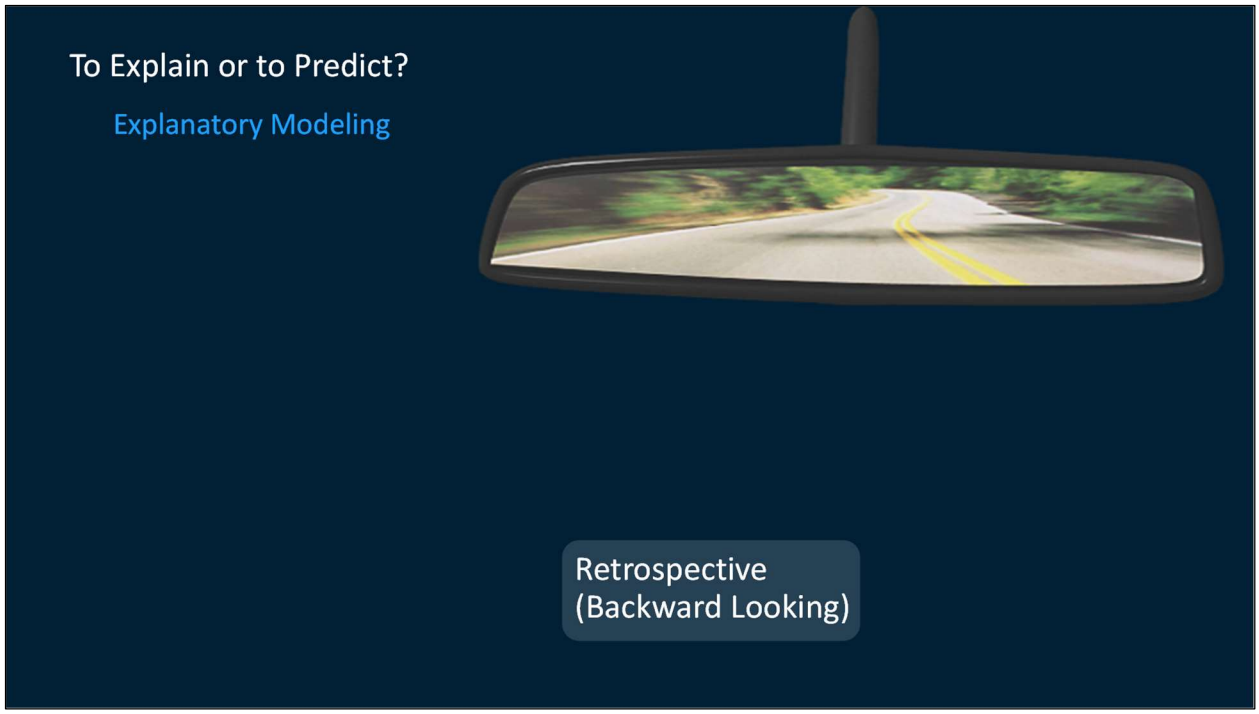
In explanatory modeling, the model represents an underlying causal function, and input is assumed to cause the target. In predictive modeling, the model captures the association between input and target.



In explanatory modeling, the model is carefully constructed based on a function in a fashion that supports interpreting the estimated relationship between input and target and testing the causal hypotheses. In predictive modeling, the model is often constructed from the data. Direct interpretability in terms of the relationship between input and target is not required, although sometimes transparency of the model is desirable.



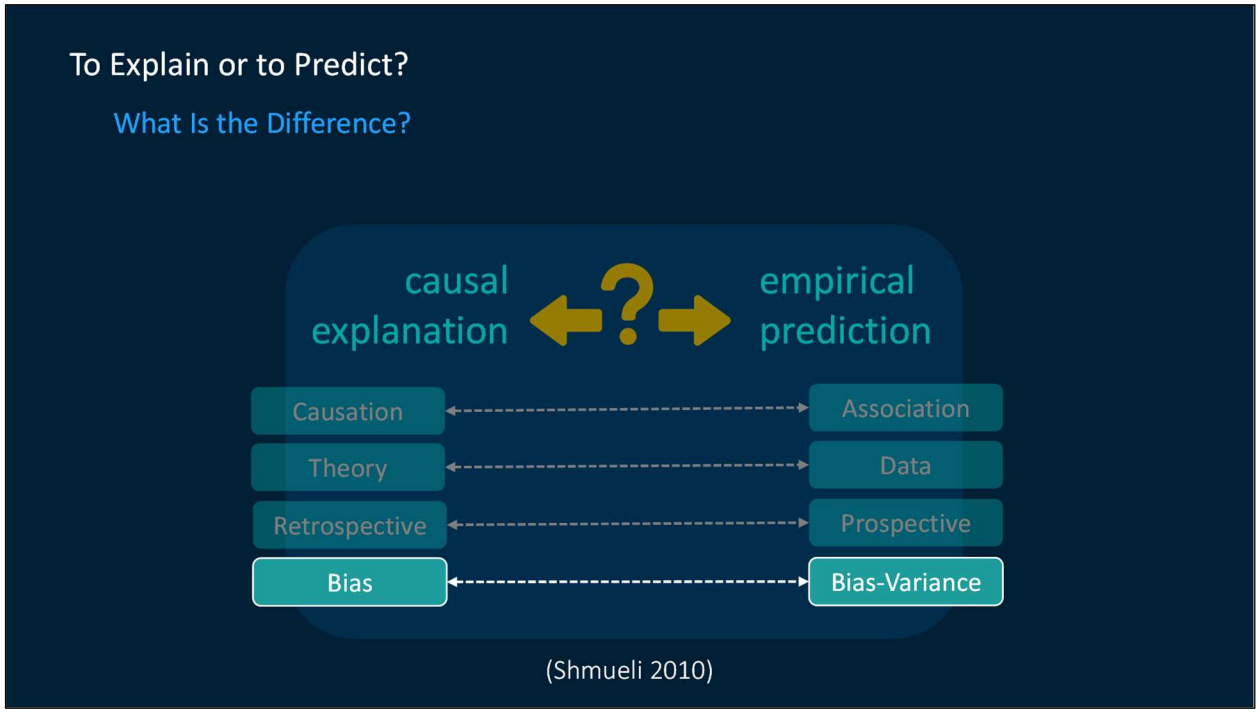
Predictive modeling is forward looking, in that the model is constructed for predicting new observations. In contrast, explanatory modeling is retrospective, in that the model is used to test an already existing set of hypotheses.



Explanatory modeling provides tools to analyze the *past*. It's retrospective, or looking backward to the past. Using explanatory modeling, you can act only on information that describes the *history*, and looking *forward* becomes a guess. So, solving your business problem based on historical data is like driving your car looking only at the rearview mirror.

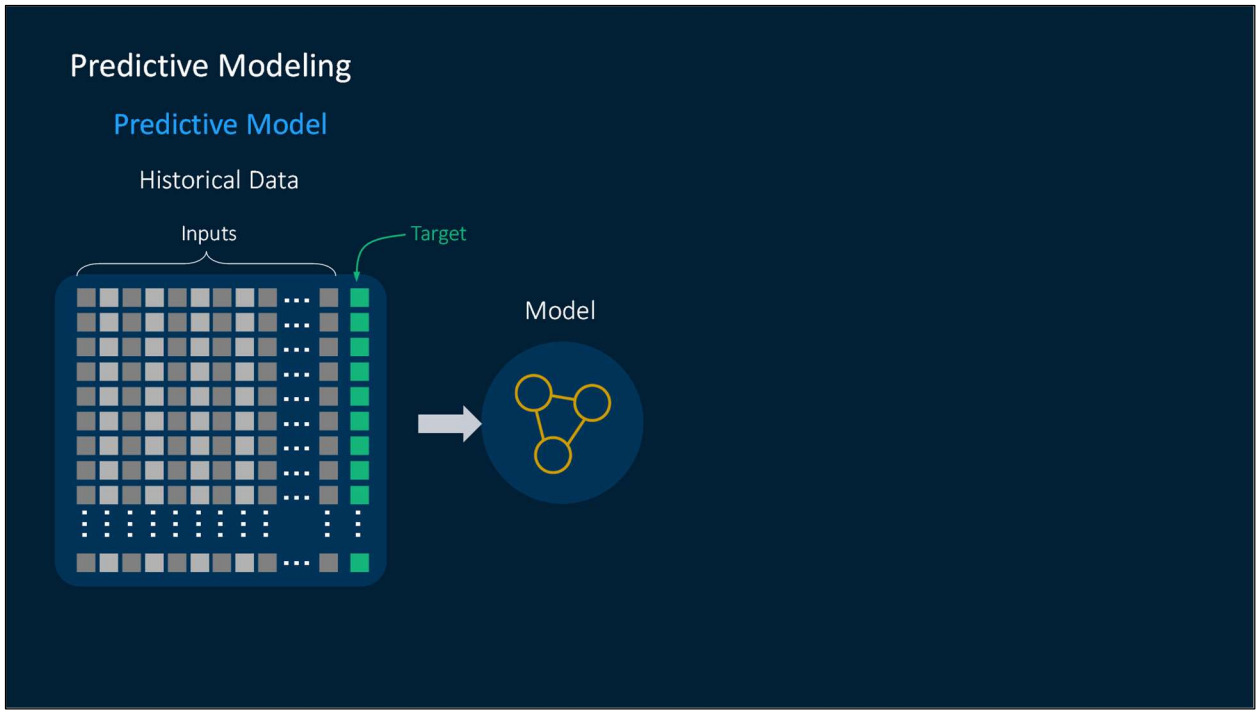
In contrast, *predictive modeling* holds the promise of being able to make actionable predictions of customer and market behavior based on historical data. Predictive modeling is prospective or

looking forward into the *future* - like seeing the truck coming toward you and being able to develop strategies to avoid the crash while also keeping eye on the rearview mirror.

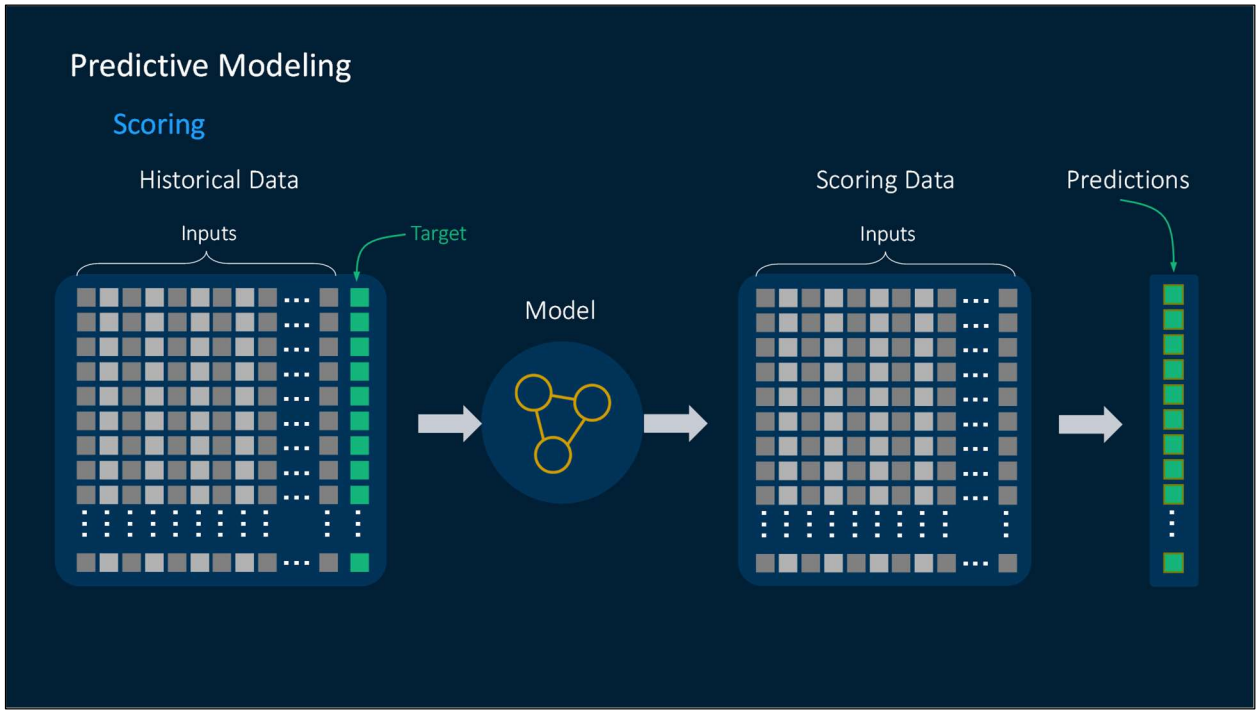


In explanatory modeling, the focus is on minimizing bias to obtain the most accurate representation of the underlying theory. In contrast, predictive modeling seeks to minimize the combination of bias and estimation variance, occasionally sacrificing theoretical accuracy for improved empirical precision.

We will discuss this bias-variance trade-off in detail in the context of predictive modeling.

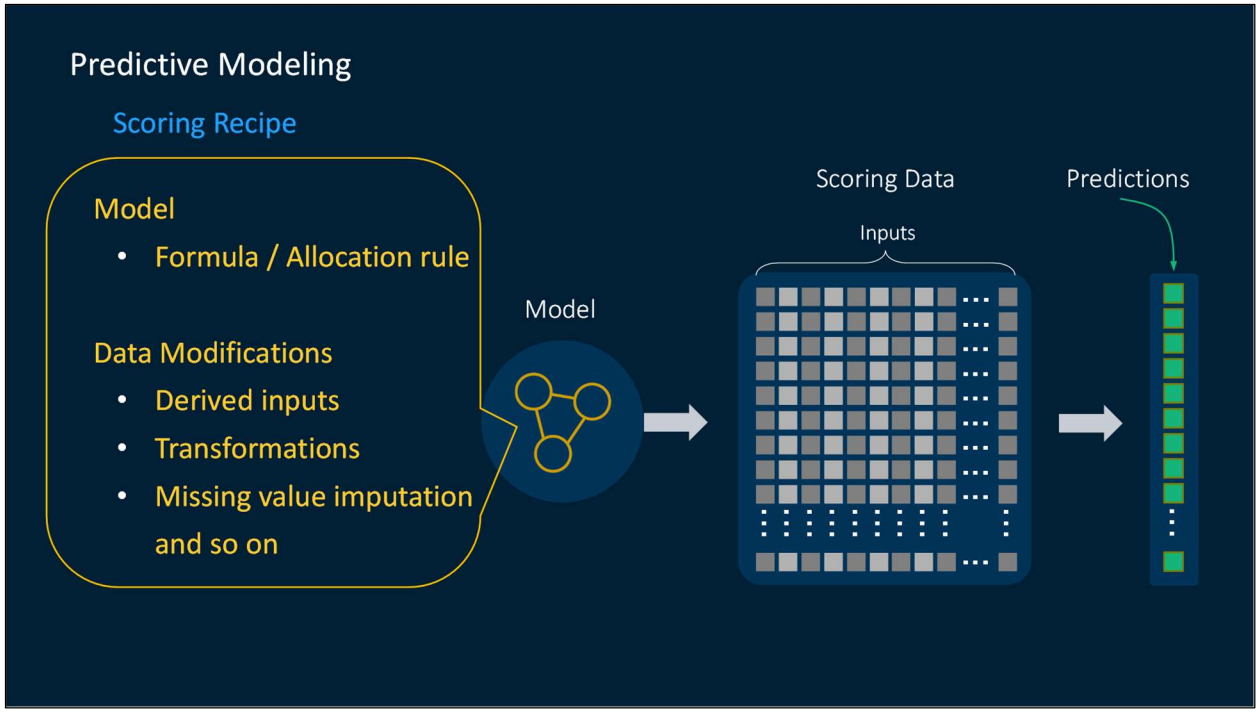


The historical data is used to construct a model (allocation rule) that can predict the values of the target from the inputs. The *predictive model* is a concise representation of the association between the inputs and the target. The task is referred to as *supervised* because the prediction model is constructed from data where the target is known.



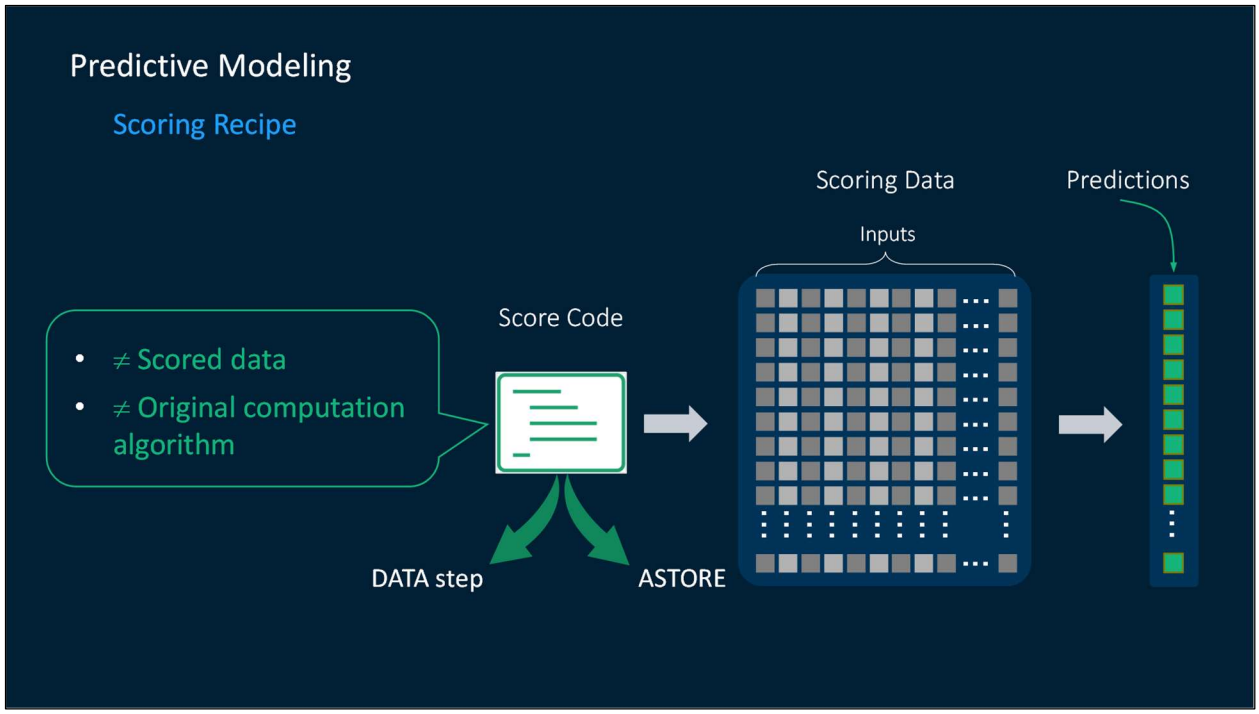
If the targets are known, why do we need a prediction model? It allows you to predict *new* cases when the target is unknown. Typically, the target is unknown because it refers to a future event. In addition, the target may be difficult, expensive, or destructive to measure.

The predictive modeling task is not completed when a model and allocation rule are determined. The model must be practically applied to new cases. This process is called *scoring*. Scoring is the generation of predicted values for a data set that might not contain a target variable. The model is applied to the scoring data set to obtain predicted outcomes. Based on the predictions from the model, the enterprise makes business decisions and acts.



The scoring process is sometimes referred to as *model deployment*, *model production*, or *model implementation*. All tasks performed earlier in the analytics life cycle led to this task: generating predictions through scoring.

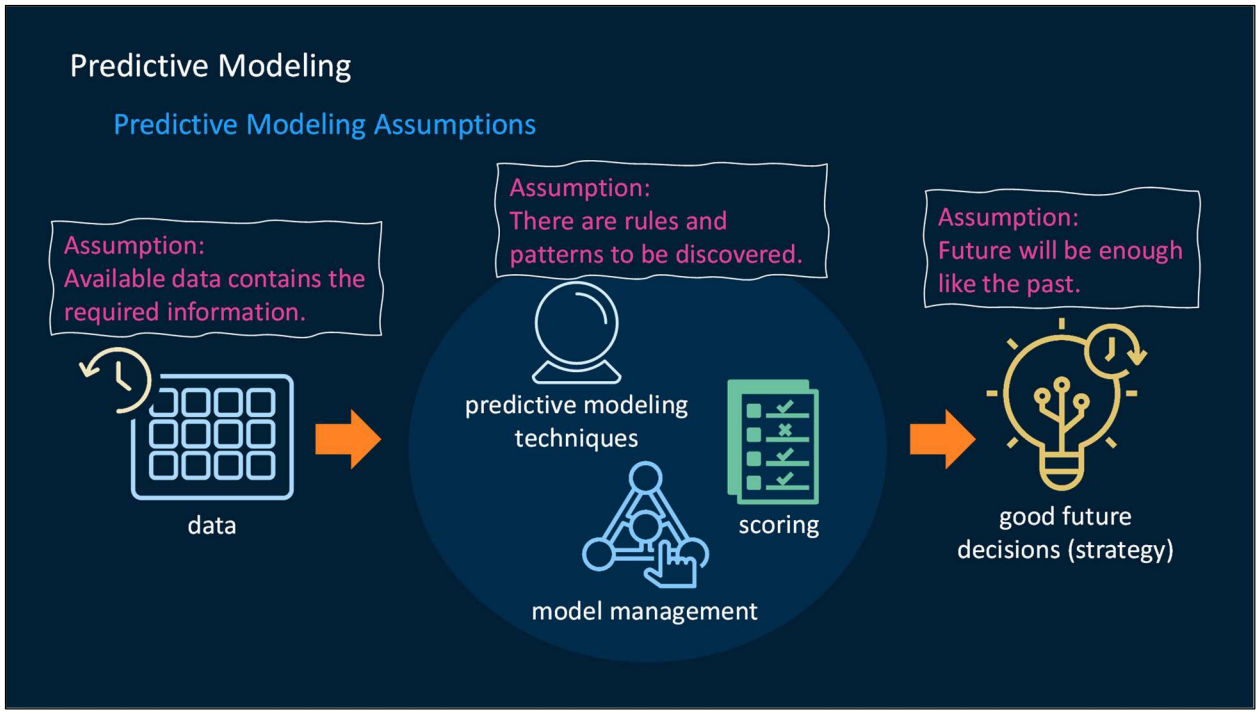
Therefore, scoring must be more than just applying the model equation or allocation rules. It must incorporate all data manipulation tasks done before generating the model like feature engineering, transformations, missing value imputation, and so on.



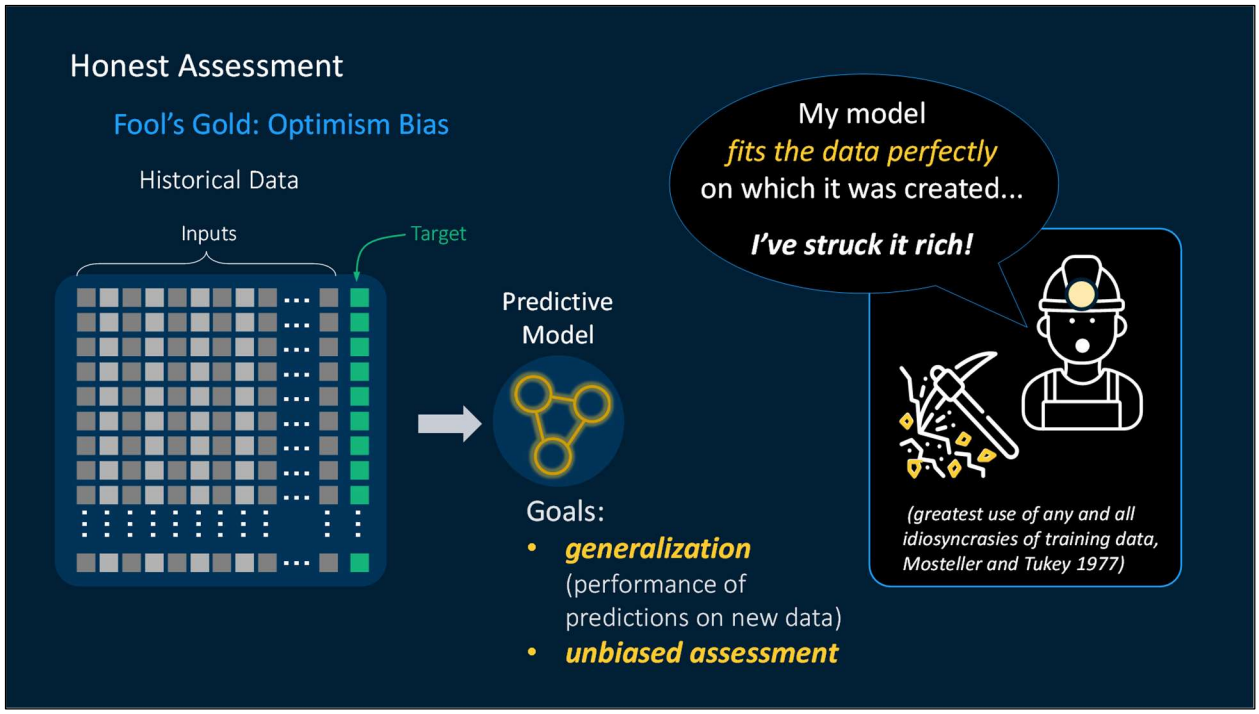
For scoring, the model is first translated into another format typically called *score code*. In the SAS Viya environment, score code is a SAS program that you can easily run on a scoring data set. Then the model is applied to the scoring data set to obtain predicted outcomes generally known as *scored data*. Based on the predictions from the model, the enterprise makes business decisions and acts.

Thus, the score code be neither confused with the scored data nor it is equivalent to the original computation algorithm.

SAS Viya creates two types of SAS language score code: either DATA step code or analytic store (ASTORE) code. An ASTORE file is a useful, transportable, universal, and compact binary file that represents the state of an analytic procedure after training.



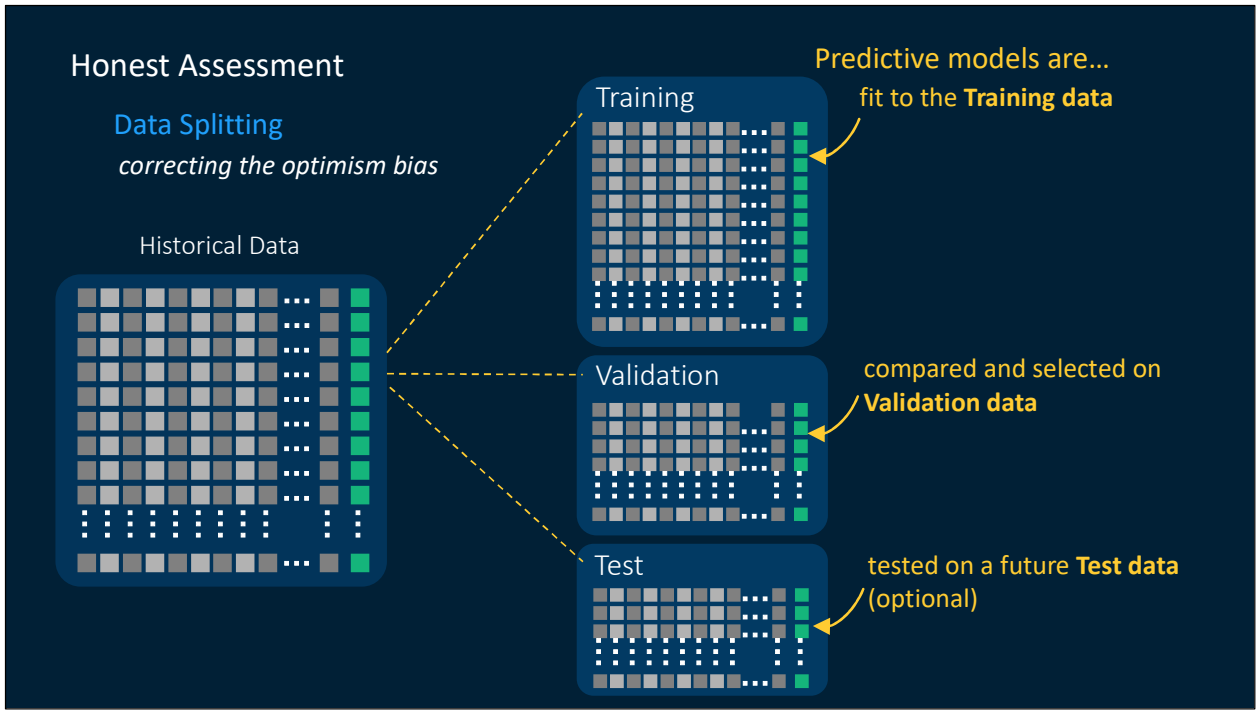
All data comes from the past. Predictive modeling techniques, paired with scoring and good model management, enable you to use your data about the past and the present to make good decisions for the future. This process assumes several things. The most important assumption is that the future will be enough like the past that lessons learned from past data will remain applicable in the future. Another important assumption is that there are rules and patterns to be discovered and that the available data contains the required information.



When you fit a predictive model on a set of historical data, you might think it's a good idea to assess the model using the same data that you used to fit the model. However, when you assess the accuracy of a predictive model on the same data that was used to fit the model, you tend to get better assessment statistics than when you assess the model on other data. This bias is known as the *optimism bias*.

The purpose of predictive modeling is *generalization*, which is the performance of the predictions on new data. Evaluating the model on the same data the model was fit on usually leads to an optimistically biased assessment.

Testing the procedure on the data that gave it birth is almost certain to overestimate performance, for the optimizing process that chose it from among many possible procedures will have made the greatest use of any and all idiosyncrasies of those particular data.



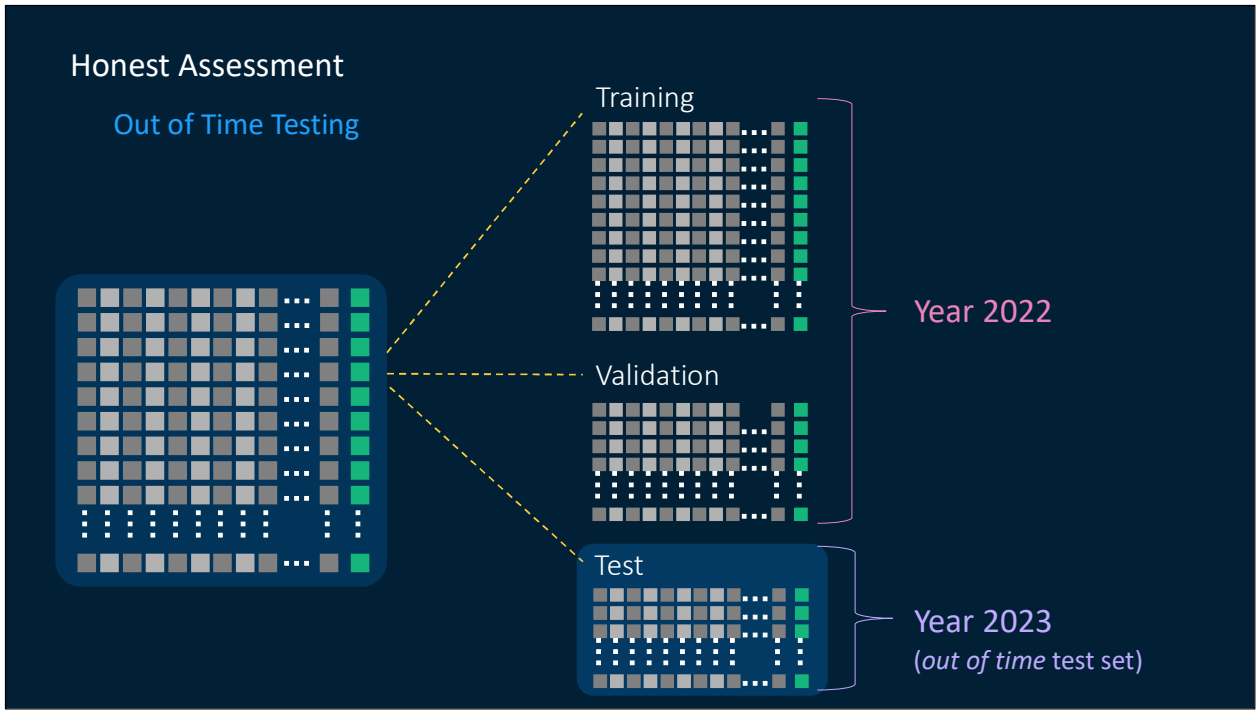
To avoid an optimistically biased assessment and create a predictive model that generalizes well, you need to assess the performance of the model on new data that was not used to fit the original model. This approach is called *honest assessment*. There are various ways to perform an honest assessment. In this lesson, you learn the simple method of splitting the data. Later in the course, you also learn about a few other methods of performing an honest assessment: k-fold cross validation and bootstrapping.

In data splitting, a portion of the data is used to fit the model and the rest is held out for empirical validation.

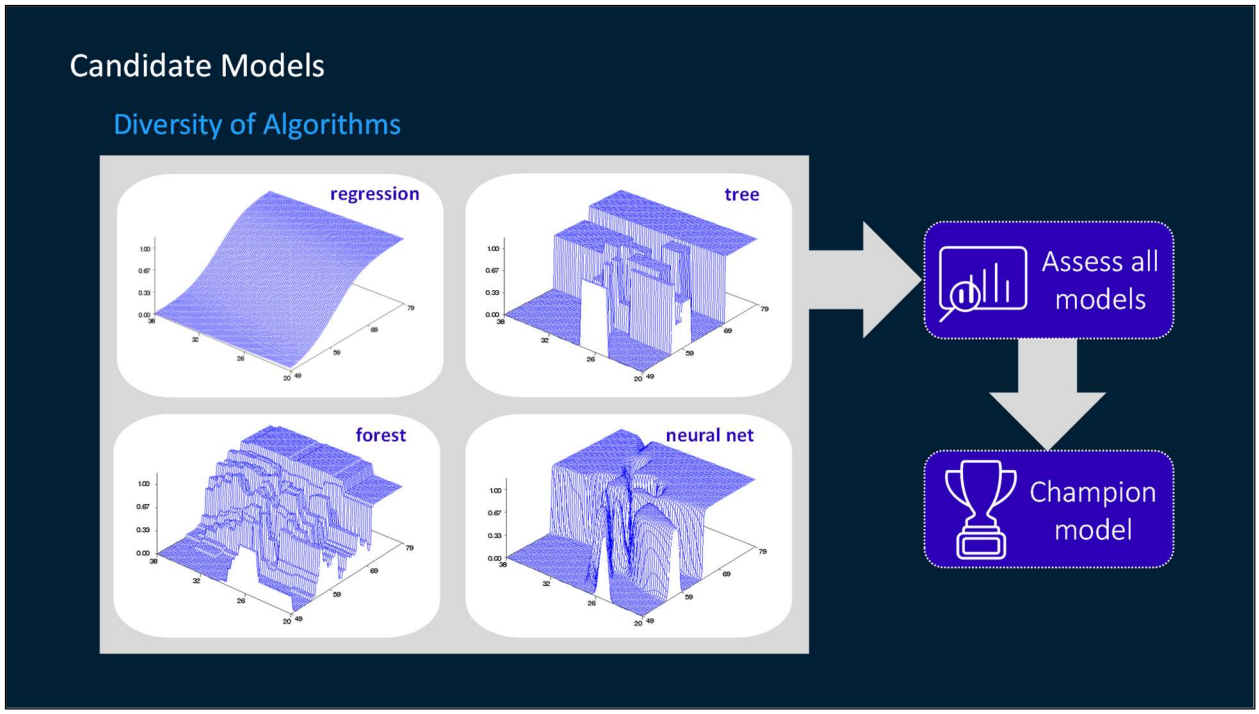
A major portion is used for fitting the model - the training data set.

The *validation data set* is used for monitoring and tuning the model to improve its generalization. The tuning process usually involves selecting among models of different types and complexities. The tuning process optimizes the selected model on the validation data. Consequently, a further holdout sample is needed for a final, unbiased assessment.

The *test data set* has only one use: to give a final honest estimate of generalization. Consequently, cases in the test set must be treated just as new data would be treated. They cannot be involved whatsoever in the determination of the fitted prediction model. In some applications, there might be no need for a final honest assessment of generalization. A model can be optimized for performance on the test set by tuning it on the validation set. It might be enough to know that the prediction model will likely give the best generalization possible without being able to say what it is. In this situation, no test set is needed.

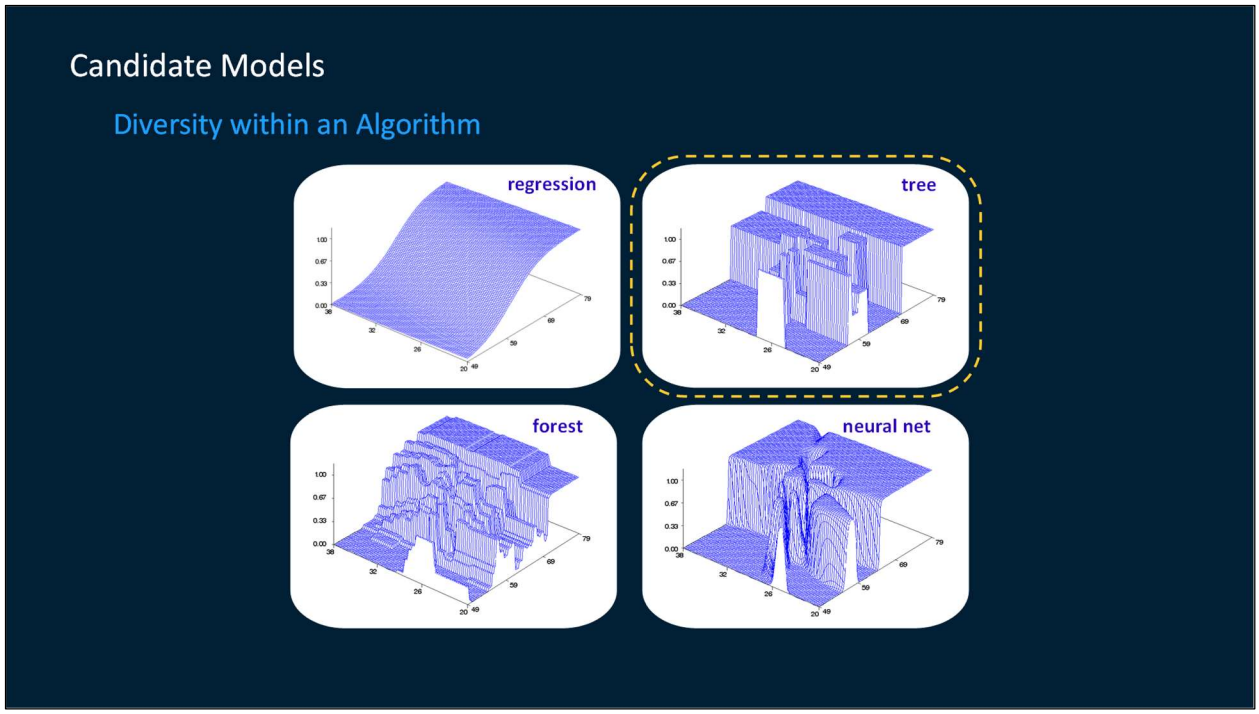


For predictive models, the test set should come from a different time period than the training and validation sets. The proof of a model's stability is in its ability to perform well month after month. A test set from a different time period, often called an *out of time* test set, is a good way to verify model stability, although such a test set is not always available.

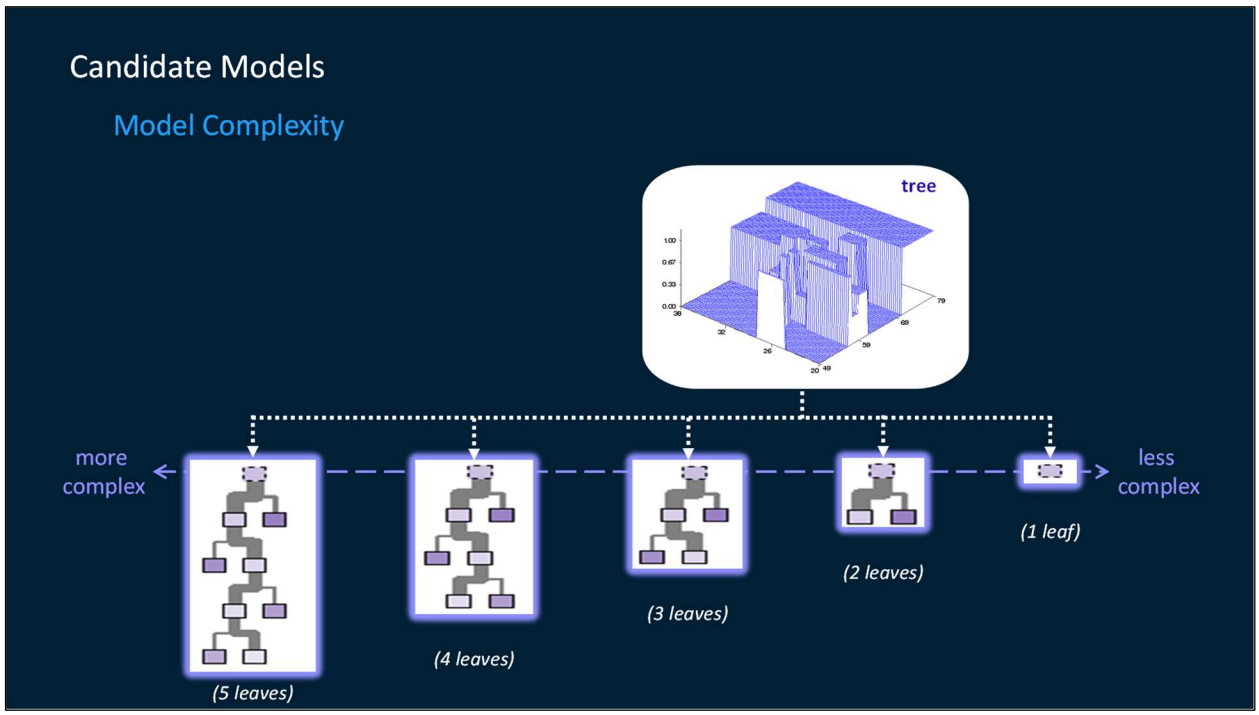


Because predictive modeling is mostly data driven, you cannot anticipate which model or algorithm works well for a given problem. Therefore, a common methodology includes building several candidate models. The models that you build can be based on diverse algorithms such as regression, decision trees, random forest, neural networks, and so on. You can see *prediction surface* plots showing how each of these four machine learning algorithms, for example, predicts a coarse grid across the input feature space.

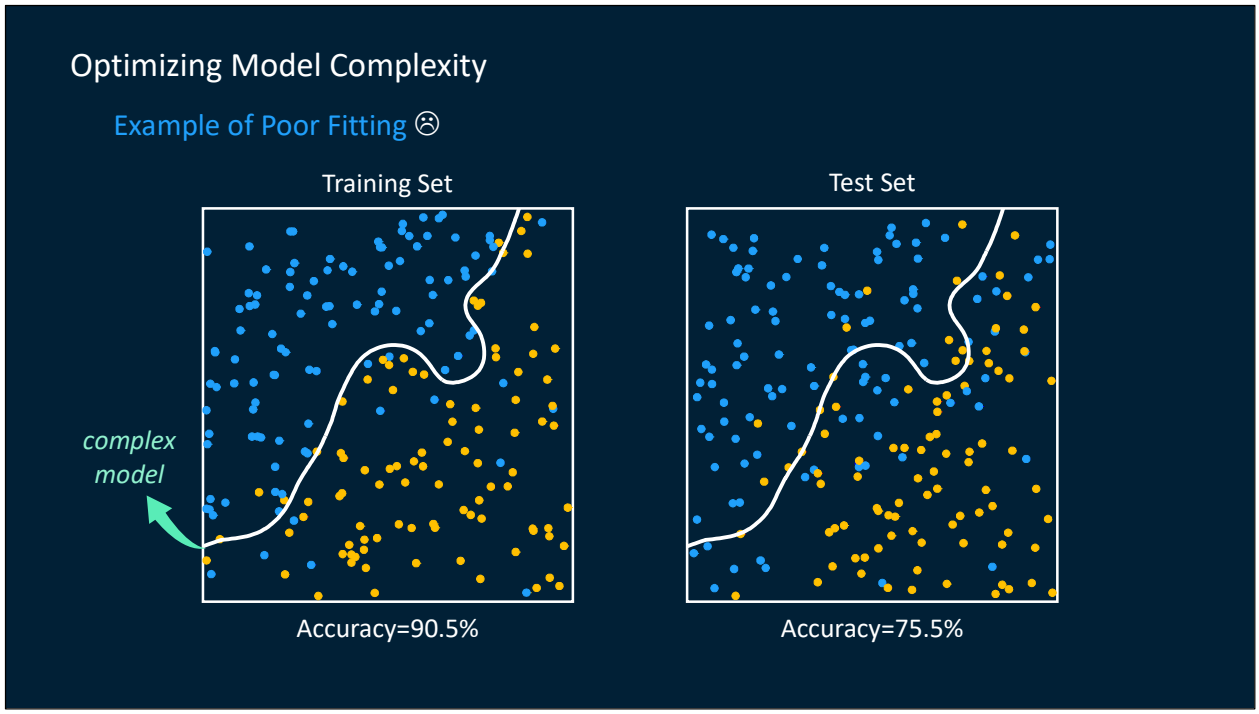
Each type of model has its own metrics by which it can be assessed, but there are also assessment tools that are independent of the type of model. By honest assessment, several statistics and statistical plots can be used to find out the best model called the champion model. The champion model is the best predictive model that is chosen from a pool of candidate models. Before you identify the champion model, you can evaluate the structure, performance, and resilience of candidate models.



Predictive modeling typically involves choices from among a set of models. These might be different *types* of models.



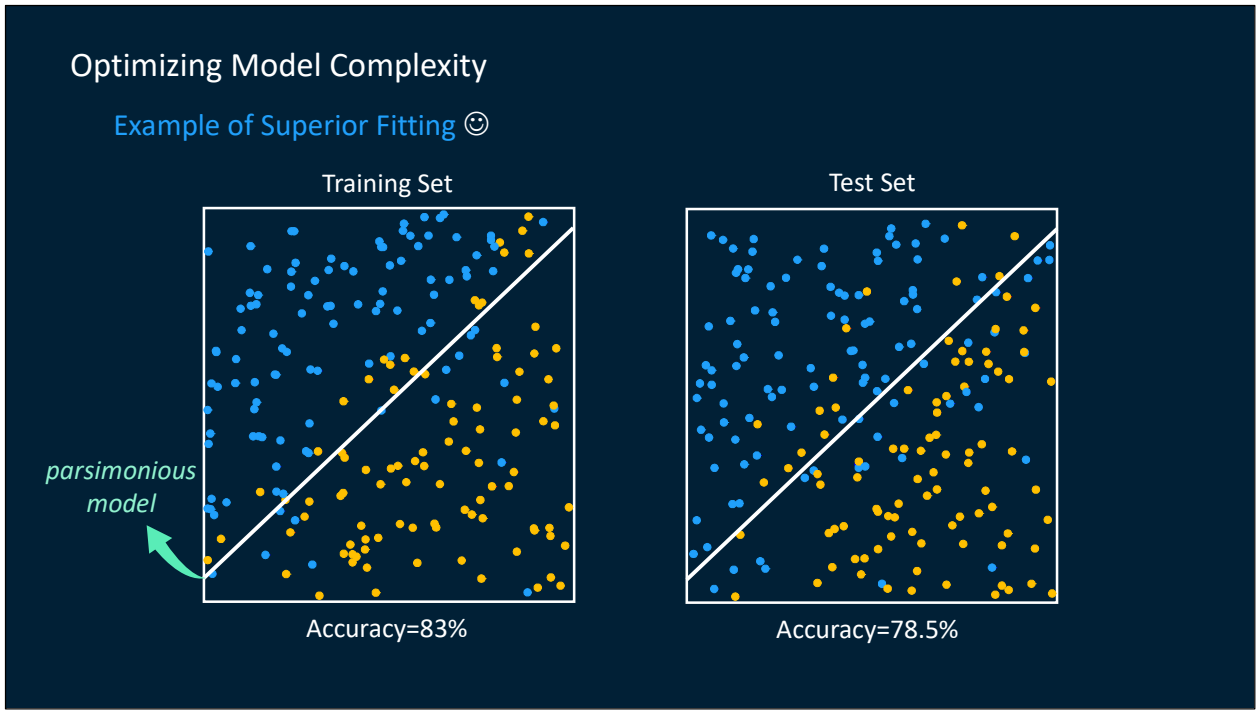
Or they might be different *complexities* of models of the *same* type, such as multiple decision trees that use different splitting and pruning criteria from the same training data set. Fitting a model to data in fact requires searching through the space of possible models. Constructing a model with good generalization requires choosing the right complexity.



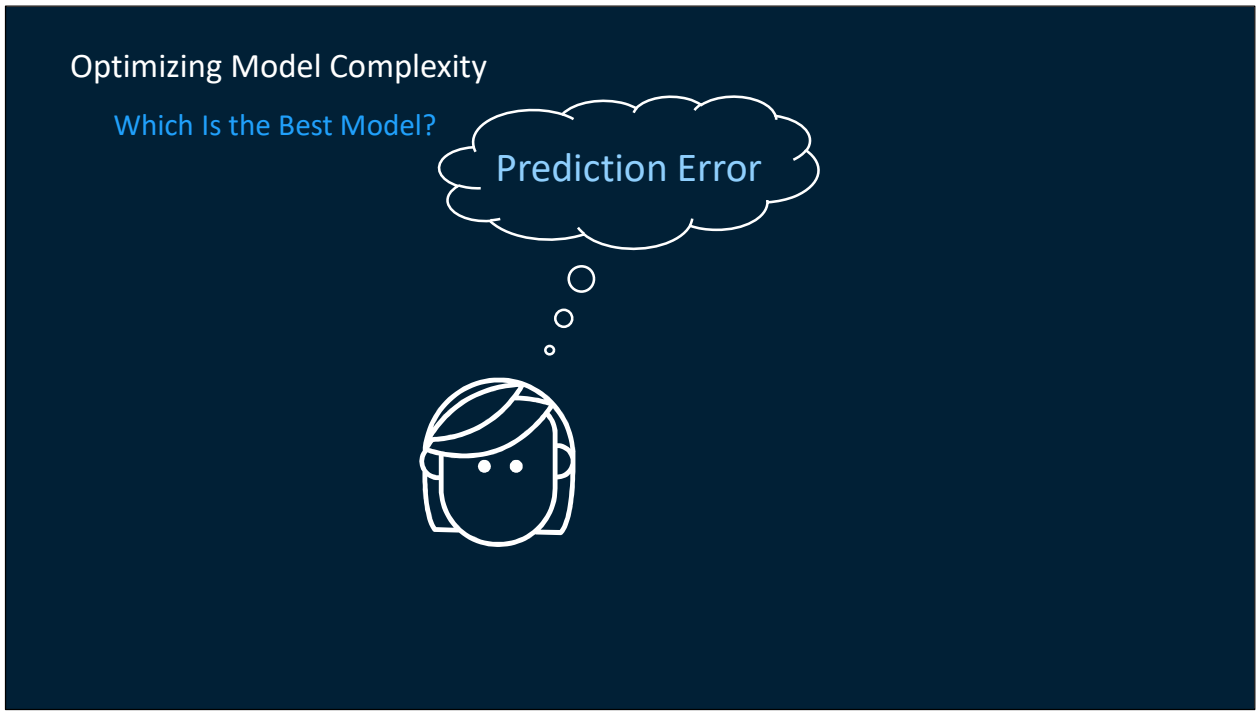
A very complex model was used on the classification problem shown here, where the goal was to discriminate between the blue and yellow classes. The classifier fit the training data well, making only 19 errors among the 200 cases (90.5% accuracy).

However, on a test set of data, the classifier did not do as well, making 49 errors among 200 cases (75.5% accuracy). The flexible model snaked through the training data accommodating the signal (meaningful information) as well as the noise (random variation or fluctuation that interferes with the signal).

Signal and noise are covered in detail later in the course.



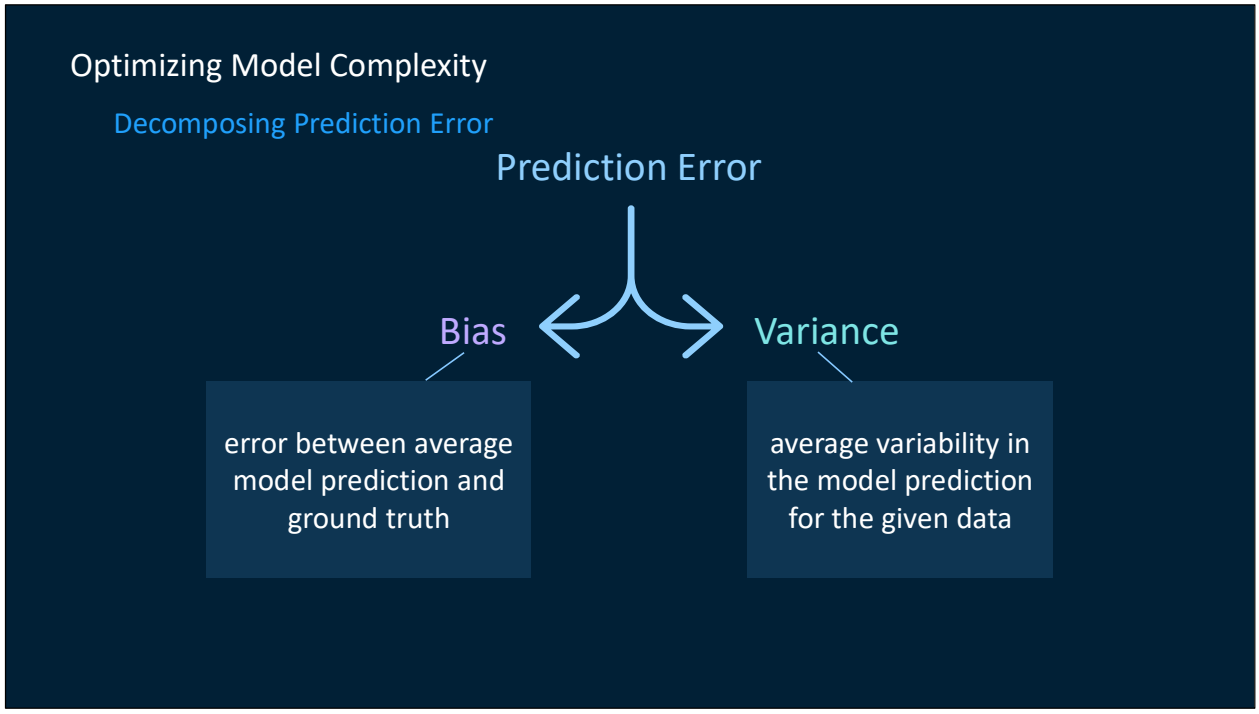
A more *parsimonious* model was fit to the training data. The apparent accuracy was not quite as impressive as the complex model (34 errors, 83% accuracy. However, it gave better performance on the test set (43 errors, 78.5% accuracy).



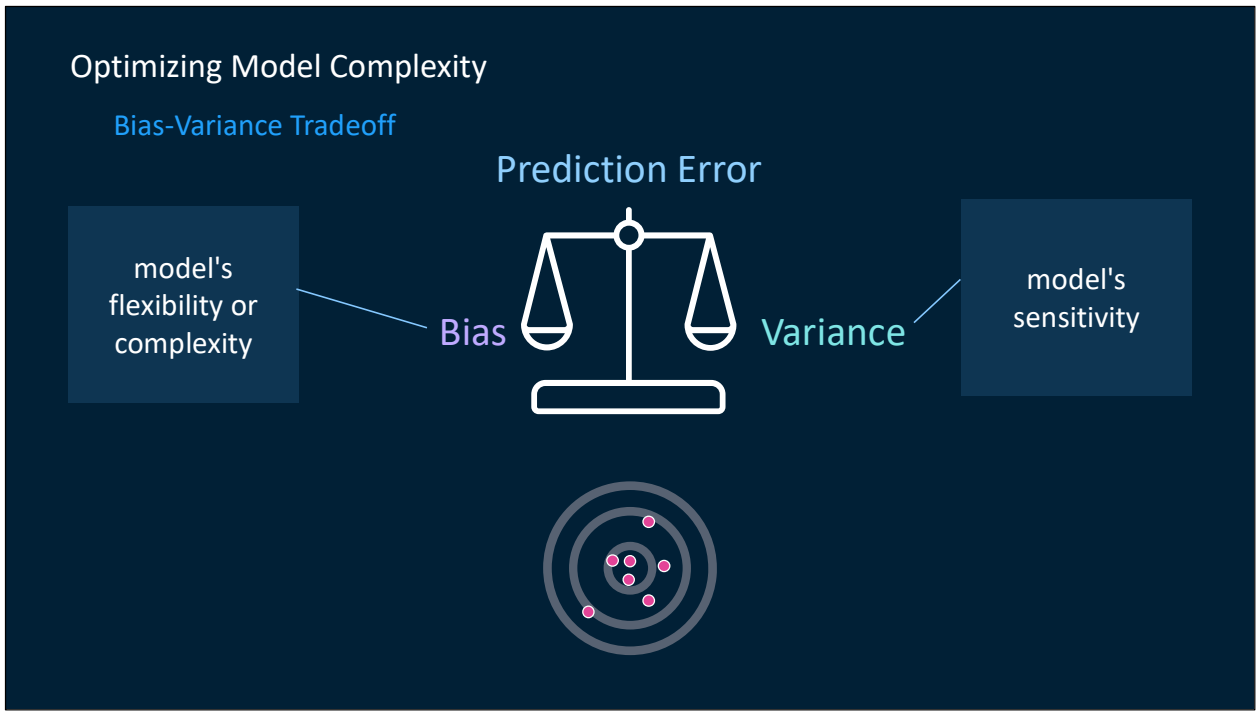
How would you decide which model to choose as best?

You can know that you picked the best model by assessing the prediction error. The prediction problem ignores questions of model correctness and focuses on the predicted values. Prediction

error refers to the difference between the predicted values made by some model and the actual values.



This expected prediction error can largely be decomposed into bias and variance. Bias is the difference between the expected prediction values and the correct value of the response. Variance is the variability of the predictions due to the model variability.

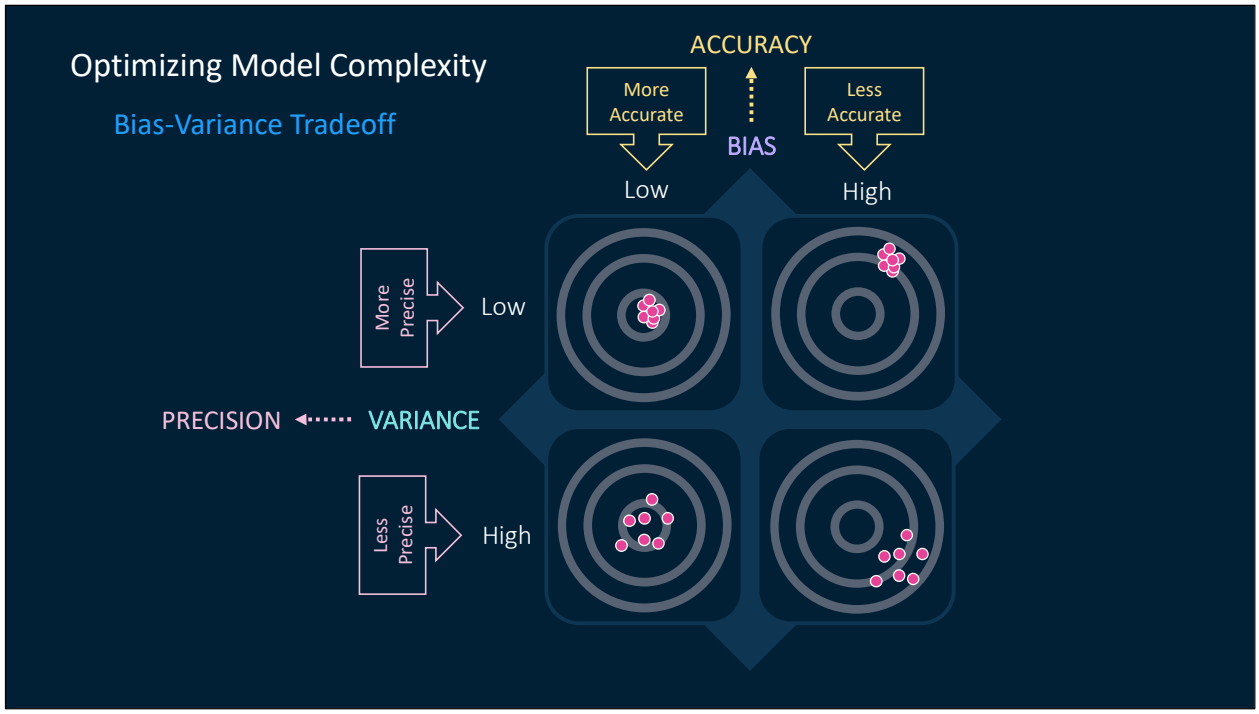


A bias occurs when an algorithm has limited flexibility to learn from data. Such models pay very little attention to the training data and oversimplify the model. The bias of the estimated function tells us the capacity of the underlying model to predict the values.

Variance defines the algorithm's sensitivity to specific sets of data. A model with a high variance pays a lot of attention to training data and overcomplicates the model that does not generalize. The variance of the estimated function tells you how much the function can adjust to the change in the data set.

Selecting model complexity involves a trade-off between bias and variance. Both bias and variance are forms of prediction error in machine learning. Thus, you select a model that simultaneously achieves *low variance* and *low bias*. But bias and variance are inversely related to each other. Trying to reduce one component, the other component of the model will increase. The true art lies in creating a good fit by balancing both.

Let's try to understand this trade-off using a simple example. Imagine that the center of the target, or the bull's-eye, perfectly predicts the correct value of your model. The dots marked on the target then represent an individual realization of your model based on your training data. In certain cases, the dots will be densely positioned close to the bull's-eye, ensuring that predictions made by the model are close to the actual data. In other cases, the training data will be scattered across the target. The more the dots deviate from the bull's-eye, the higher the bias and the less accurate the model will be in its overall predictive ability.



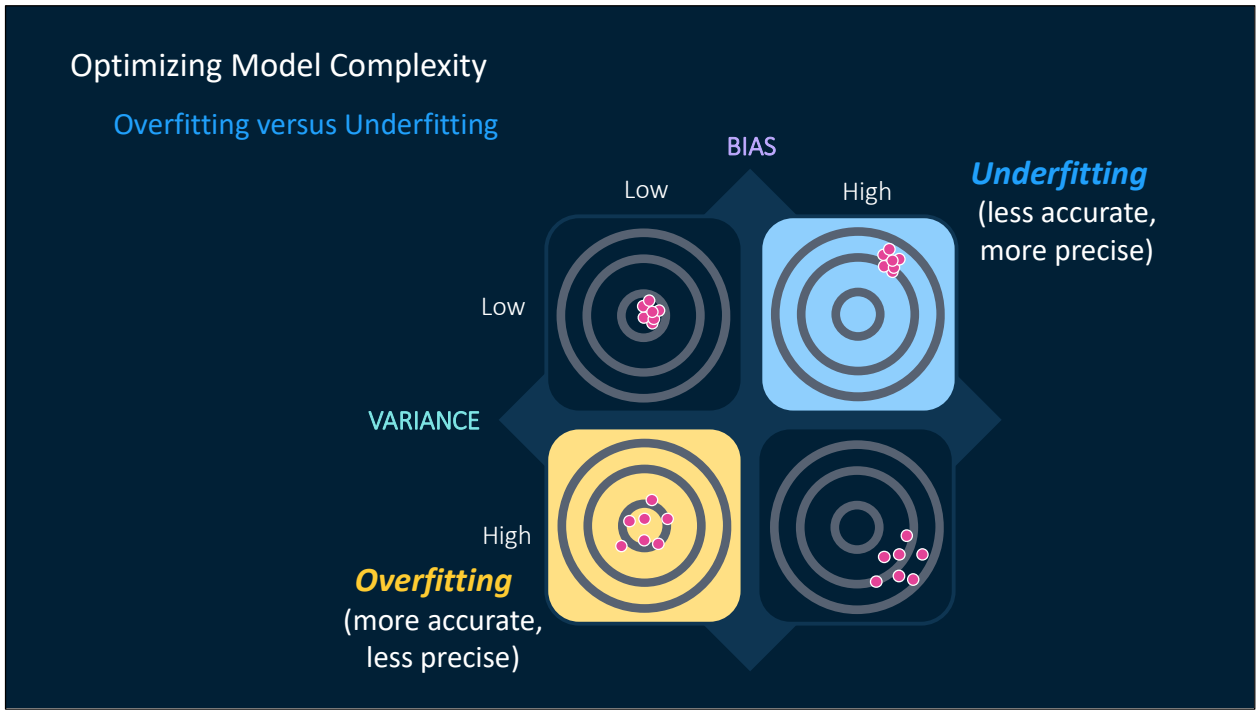
In the first target (located on the top left) , we can see an example of low bias and low variance. Bias is low because the hits are closely aligned to the center and there is low variance because the hits are densely positioned in one location.

The second target (located on the bottom left) shows a case of low bias and high variance. Although the hits are not as close to the bull's-eye as in the previous example, they are still close to the center, and bias is therefore relatively low. However, there is high variance this time because the hits are spread out from each other.

The third target (located on the top right) represents high bias and low variance, and the fourth target (located on the bottom right) shows high bias and high variance.

Hence, you can never make exact measurements in an experiment. You often use accuracy and precision for assessing predictive models. How far away you are from the "mark" is described by *accuracy,* and how well you measure is described by *precision*. An analogy can be made to the relationship between accuracy and precision. Accuracy is a description of bias. Low bias corresponds to more accurate. Conversely, high bias corresponds to less accurate. Accuracy is measuring near the target/true value.

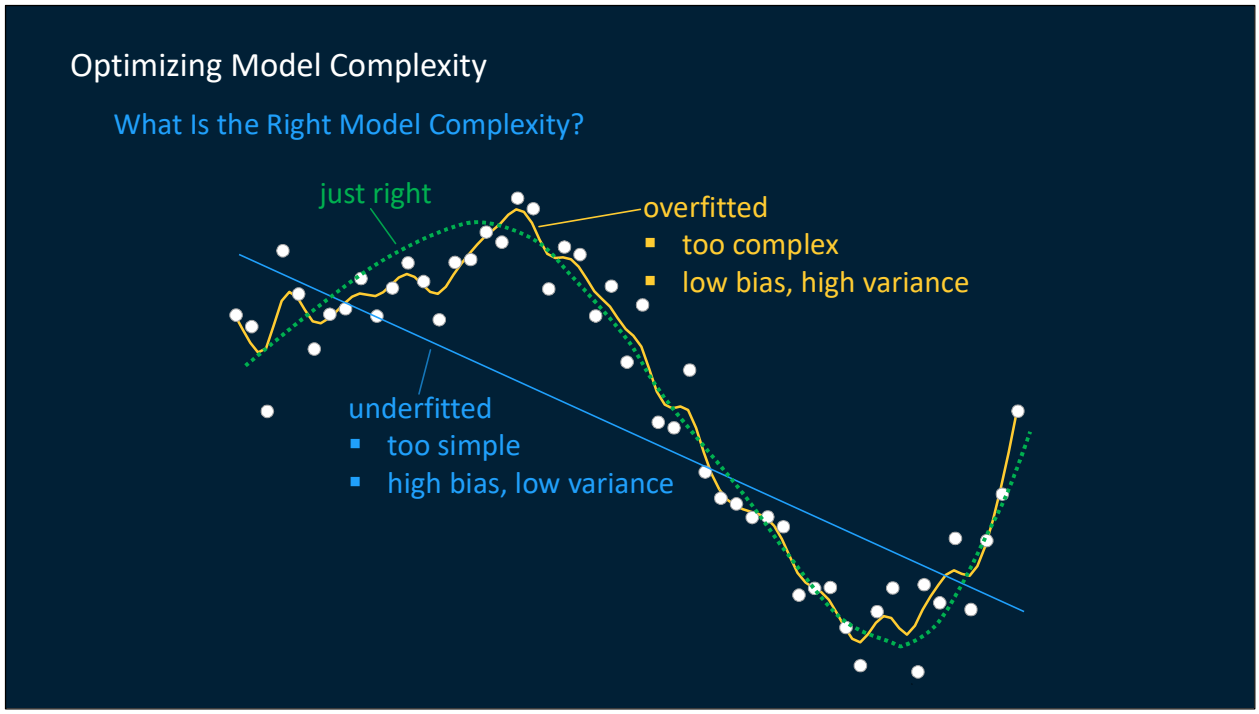
Precision is a description of variance. Low variance corresponds to more precise and whereas high variance corresponds to less precise. Precision is getting consistent results of repeated measurements.



Underfitting happens when a model is unable to capture the underlying pattern of the data. These models usually have high bias and low variance. In other words, they are less accurate and more precise.

Overfitting happens when a model captures the noise along with the underlying pattern in data. These models have low bias and high variance. In other words, they are more accurate and less precise.

A model with low bias and low variance is desirable, and a model with high bias and high variance is not.

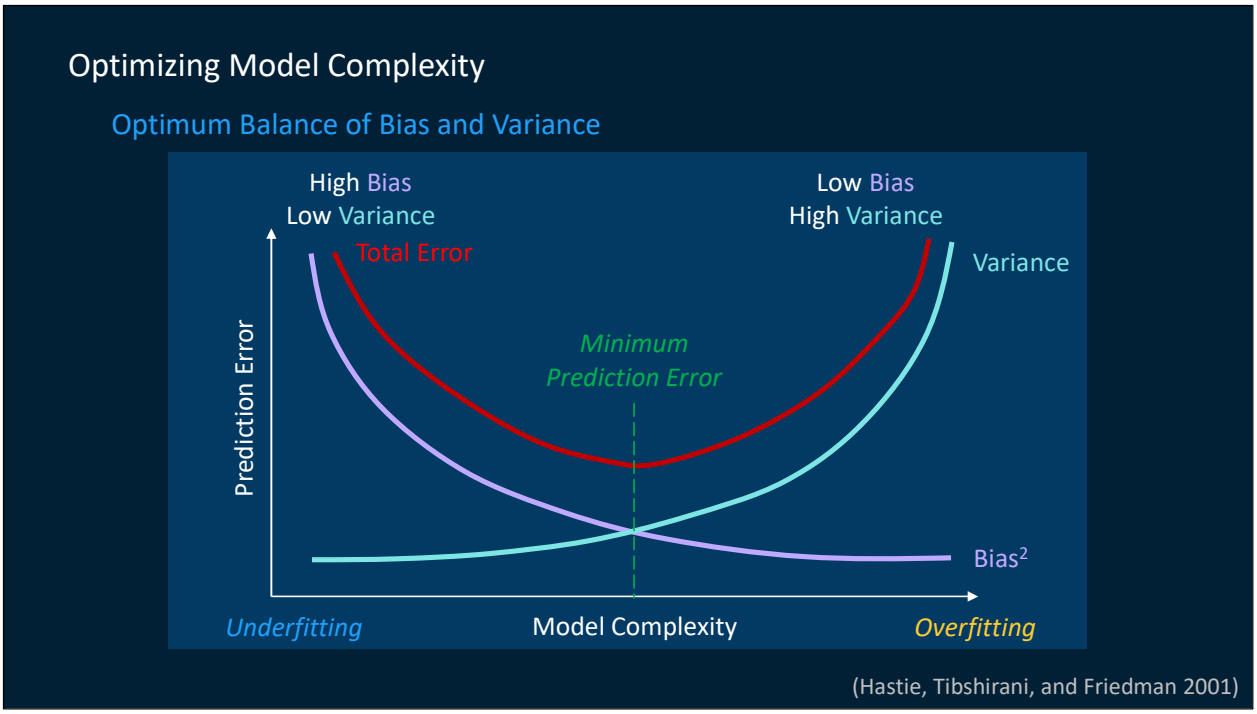


Fitting a model to data requires searching through the space of possible models. Constructing a model with good generalization requires choosing the right complexity.

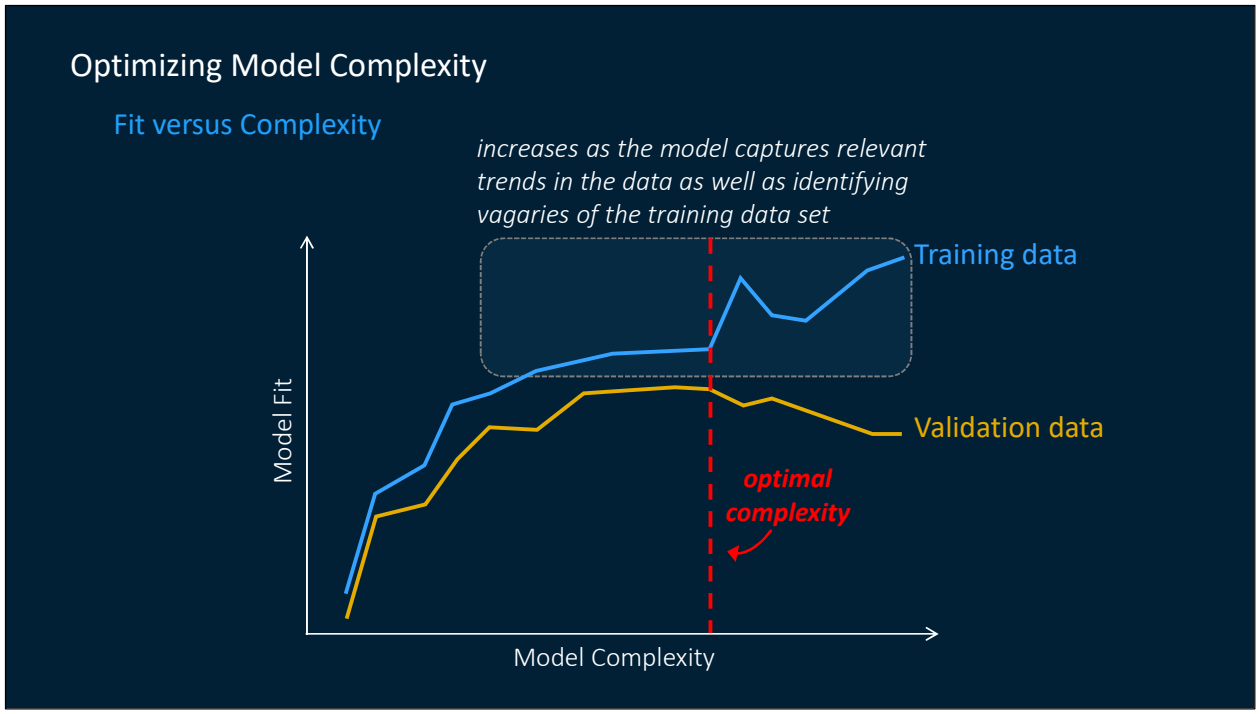
An insufficiently complex model might not be flexible enough. This leads to underfitting – systematically missing the signal. Simple, less complex models do not have the flexibility to fit every data set that well. They do not approximate the true relationship between predictors and the response variable.

An overly complex model might be too flexible. This will lead to overfitting – accommodating nuances of the random noise in the sample. These models, due to their flexibility, can fit training data too much, creating poor prediction performance in a new data set.

A model with just enough flexibility will give the best generalization. The strategy for choosing model complexity in data mining and machine learning is to select the model that performs best on the validation data set. Using the training data to evaluate performance usually leads to selecting too complex a model. (The classic example of this is selecting linear regression models based on R^2.



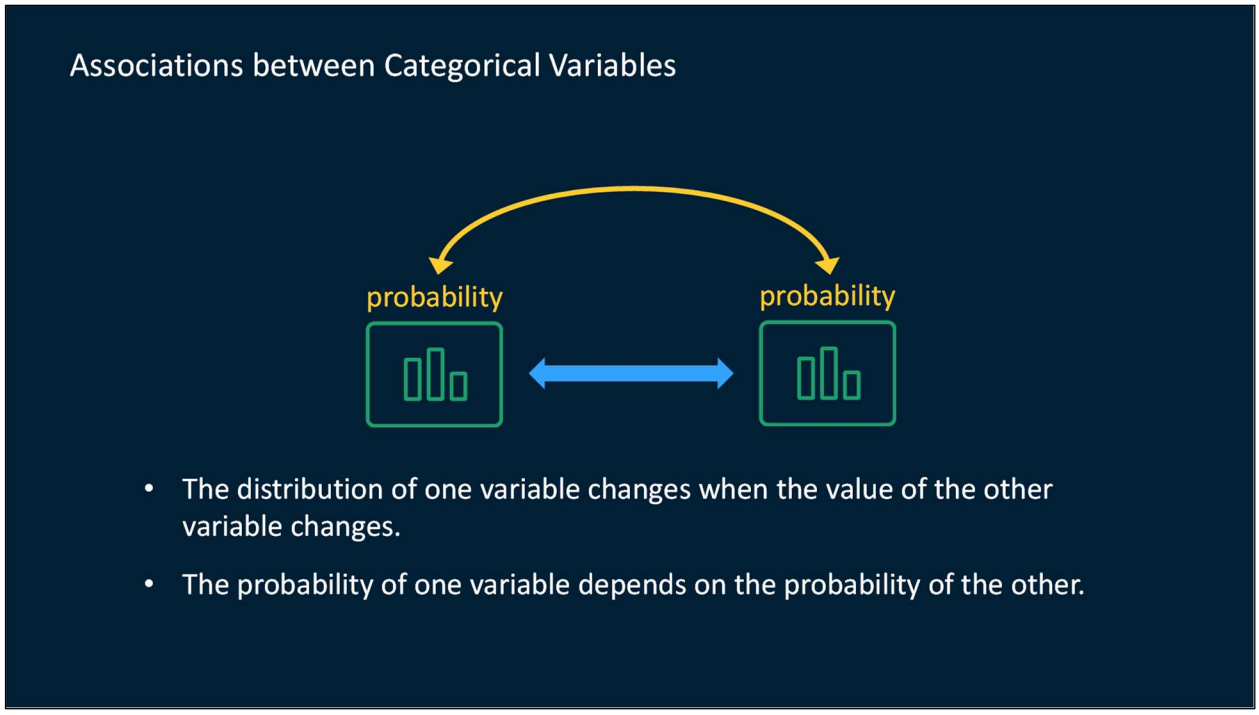
In reality, there is always a trade-off between bias and variance. Bias and variance both contribute to error, but it is the prediction error that you want to minimize, not bias or variance specifically. An optimal balance of bias and variance would never overfit or underfit the model. Increasing the variance with complex models will decrease the bias, but that might overfit the model. Conversely, simple models will increase the bias at the expense of the model variance, and that might underfit the model. There is an optimal point for model complexity, a balance between overfitting and underfitting. In practice, there is no analytical way to find this optimal complexity. Instead, we must use an accurate measure of prediction error, explore different levels of model complexity, and choose the complexity level that minimizes the overall error.



To compare many models, an appropriate fit statistic must be selected. For a series of models with increasing complexity, which could be generated by an automatic selection routine, it is conceivable to plot a fit measure against some index of complexity.

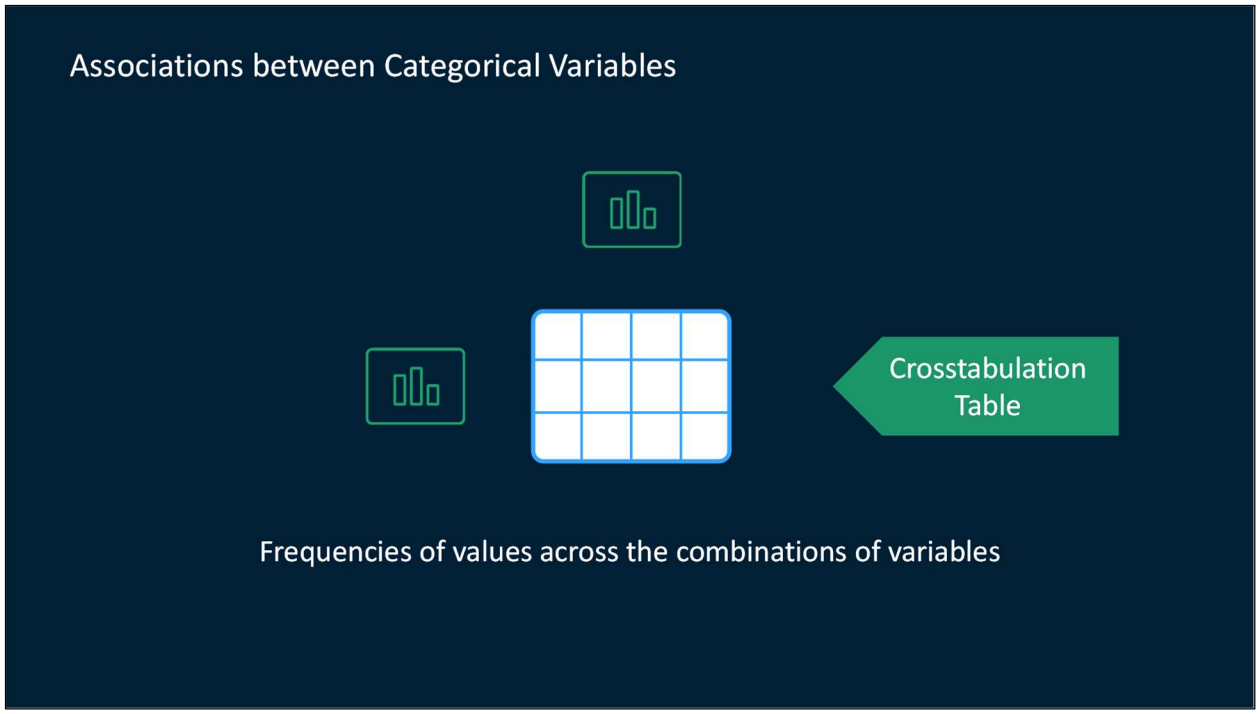
Typically, model performance follows a straightforward trend. As the complexity increases the fit on the training data gets better. After a point, the fit might plateau, but on the training data, the fit gets better as model complexity increases. Some of this increase is attributable to the model capturing relevant trends in the data. Detecting these trends is the goal of modeling. Some of the increase, however, is due to the model identifying vagaries of the training data set. This behavior has been called overfitting. Because these vagaries are not likely to be repeated, in the validation data or in future observations, it is reasonable to want to eliminate those models. Hence, the model fit on the validation data, for models of varying complexity, is also plotted. The typical behavior of the validation fit line is an increase (as more complex models detect more usable patterns) followed by a plateau, which might finally result in a decline in performance. The decline in performance is due to overfitting. The plateau just indicates more complicated models that have no fit-based arguments for their use. A reasonable rule would be to select the model associated with the complexity that has the highest validation fit statistic.

4.2 Categorical Associations



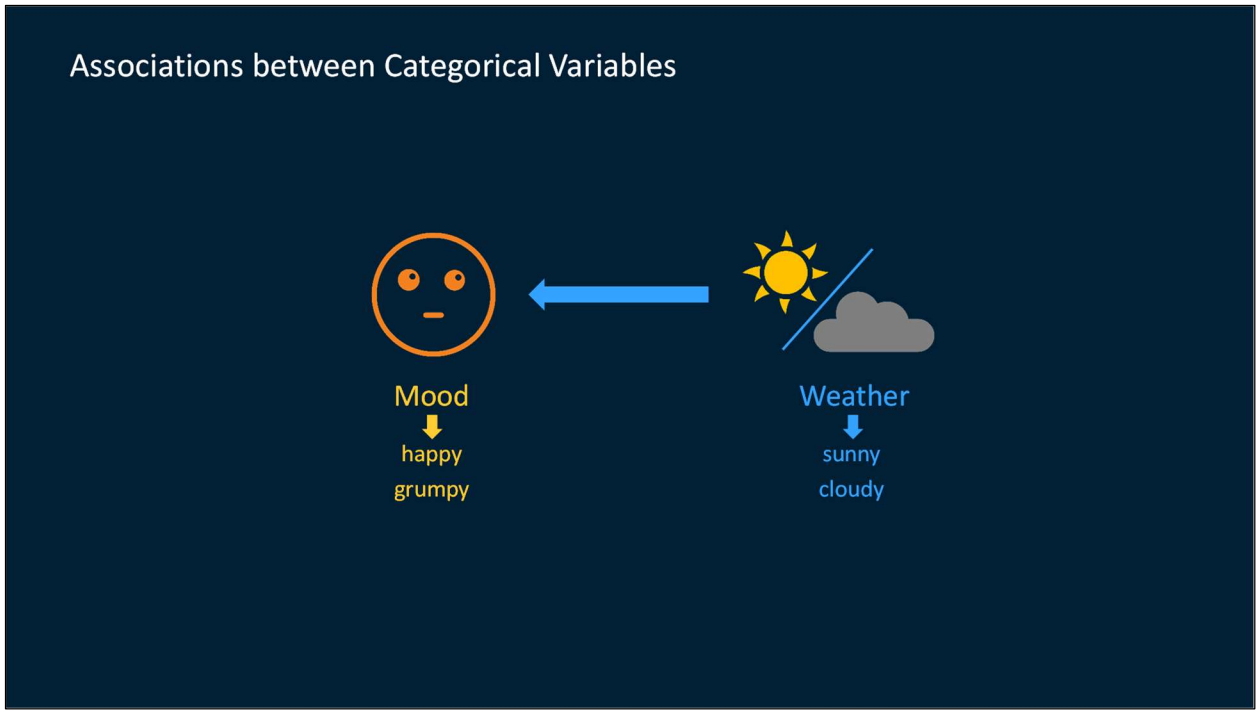
By examining distributions of categorical variables, you can determine the frequencies of data values and possible associations among variables. An association exists between two categorical variables if the distribution of one variable changes when the value of the other variable changes.

You can also think of an association as the probability that one variable depends on the probability of the other. More precisely, the difference between conditional and marginal probabilities is an indication that the variables might be associated. If there's no association, the distribution of the first variable is the same, regardless of the level of the other variable.

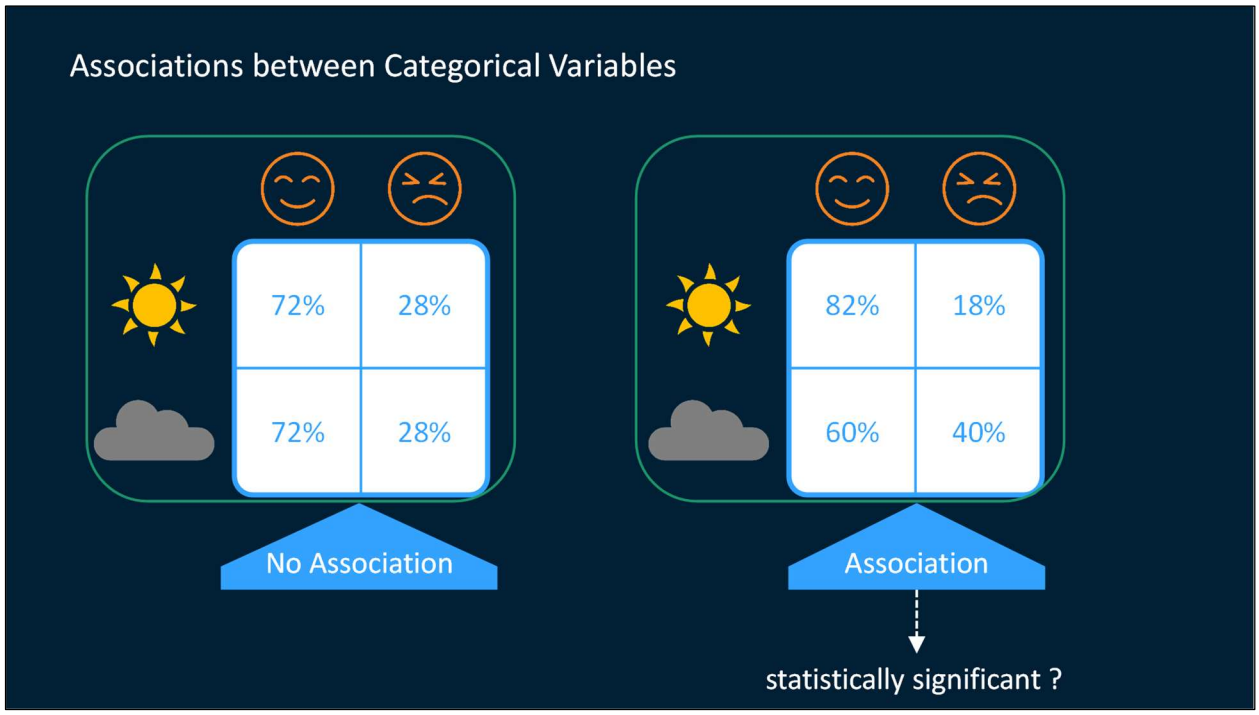


To look for associations, you examine the frequencies of values across the combinations of variables.

Crosstabulation is commonly used to quantitatively analyze the relationship between two or more categorical variables. Cross tabulations — also referred to as contingency tables or crosstabs — group categorical variables together and enable you to understand their association.



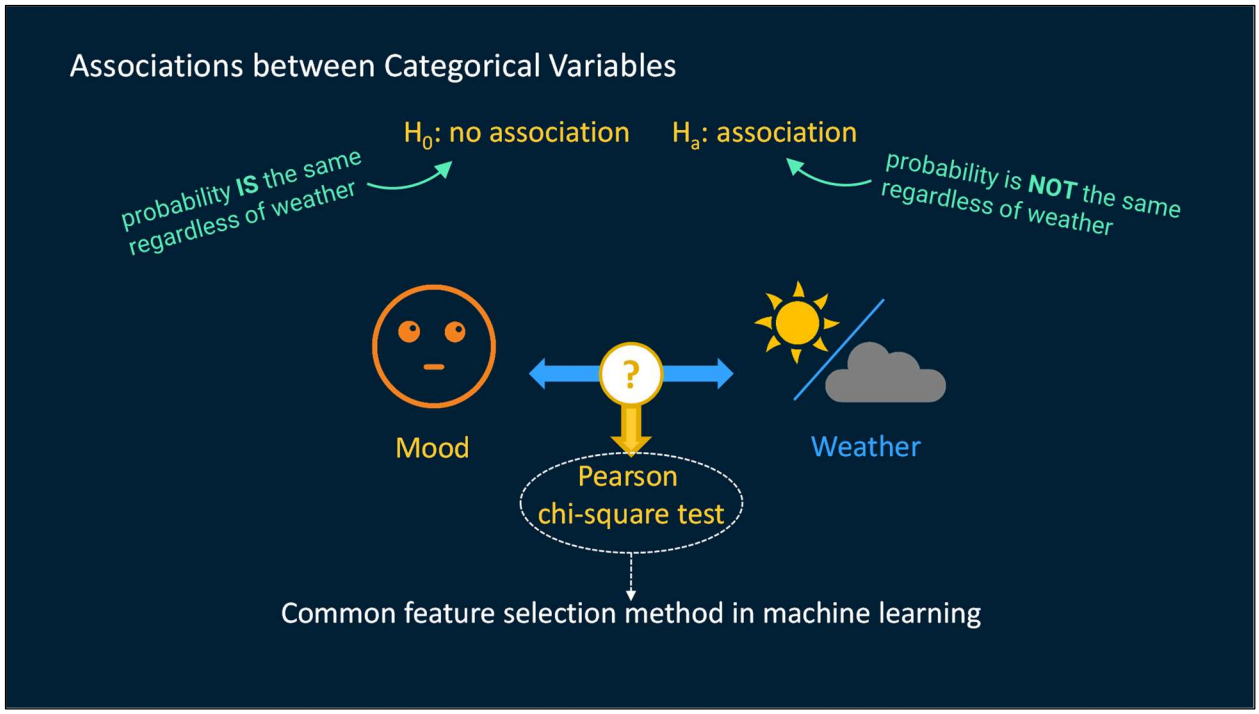
For example, suppose you want to determine whether your mood is affected by the weather. The categorical response variable is **Mood**, and its values are either *happy* or *grumpy*. The categorical predictor variable is **Weather**, and its values are either *sunny* or *cloudy*.



This table shows row frequency percentages for combinations of values of the two variables. On sunny days, you're happy 72% of the time and grumpy 28% of the time. Now look at the frequency percentages for your mood on cloudy days. The row percentages are the same in each column, indicating that there's no change in your mood based on the weather. So, there's no association between these two variables.

What if these were your results? In this table, the row percentages are different in each column. On sunny days, you're happy 82% of the time and grumpy 18% of the time. On cloudy days, you're happy 60% of the time and grumpy 40% of the time. Your mood changes based on the weather. It appears you're more likely to be happy if the weather is nicer, indicating a possible association between mood and weather.

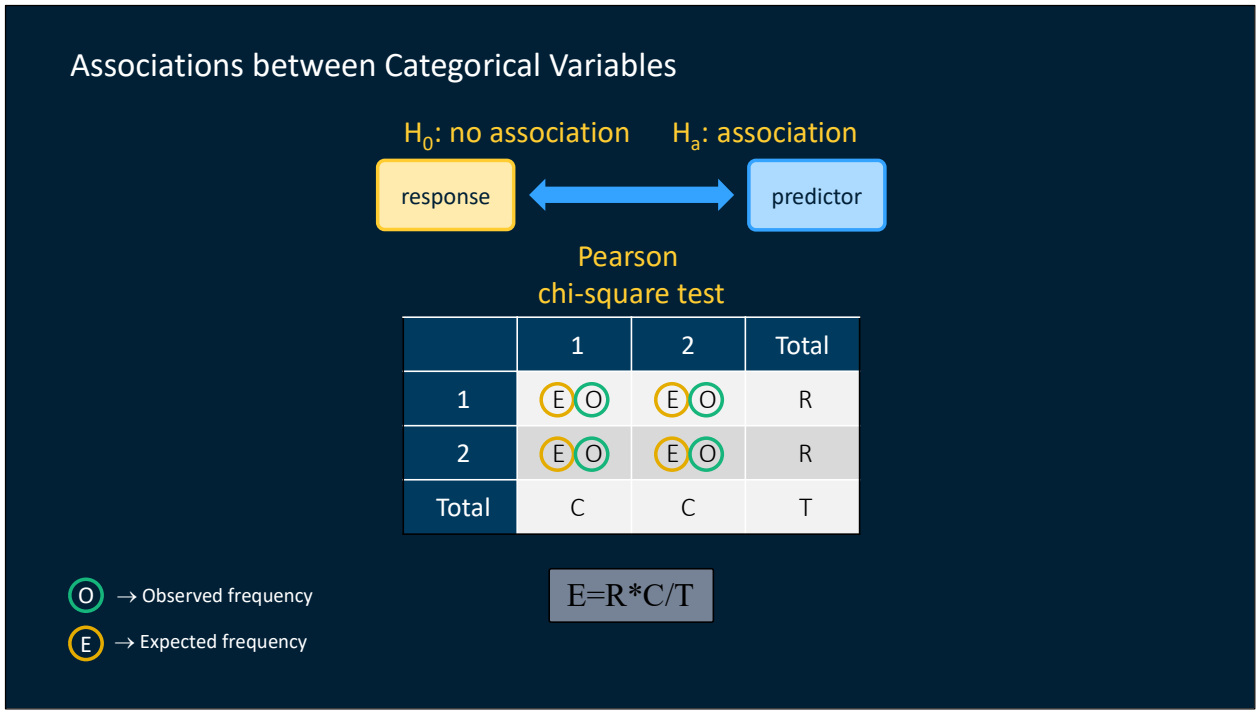
We need to assess whether the differences between the percentages of **Mood** across levels of **Weather** are greater than would be expected by chance. To be certain that the variables are associated, we need to run a formal test of association, the chi-square test.



Let's start with our null hypothesis, that there's no association between the variables **Weather** and **Mood**, meaning that the probability of a happy mood is the same regardless of weather.

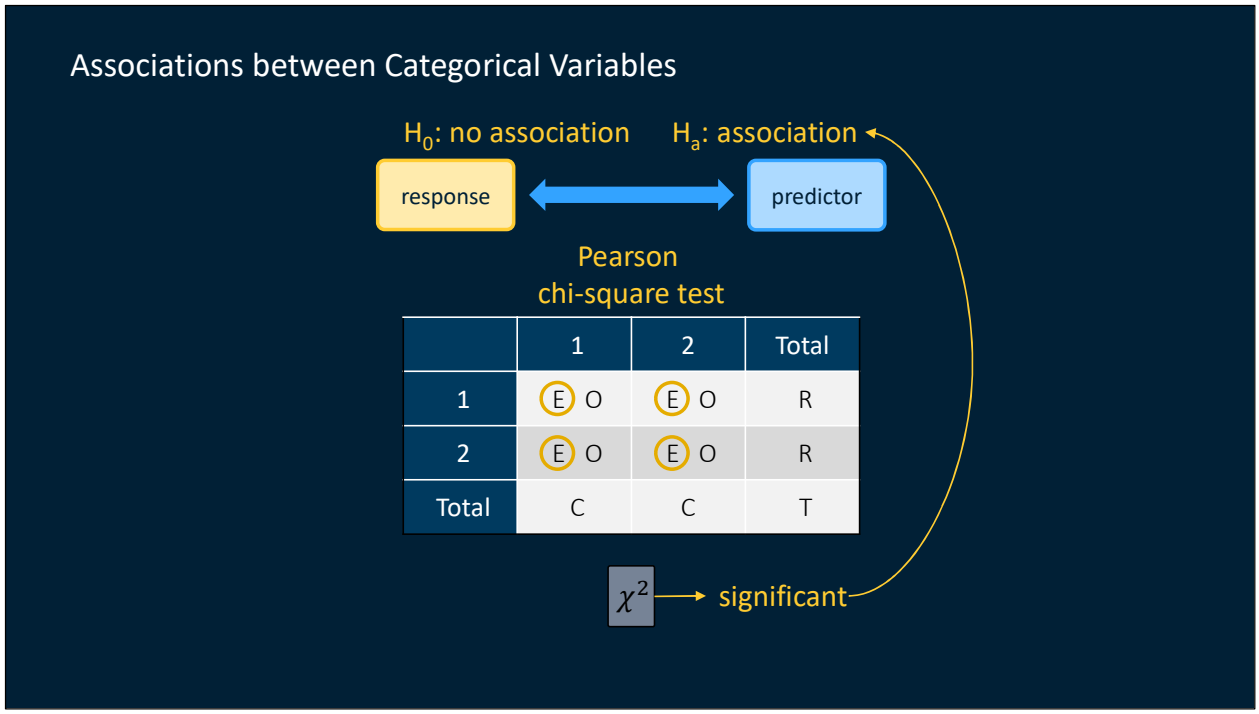
The alternative hypothesis is that there *is* an association between **Weather** and **Mood**, meaning the probability of a happy mood is *not* the same for sunny and cloudy weather.

To formally test the association for statistical significance, we'll use the Pearson chi-square test, often referred to as simply the chi-square test. Chi-square test is commonly used for feature selection in machine learning. For example, it can be used as a split search criterion in decision trees. In feature selection, we aim to select the features that are highly dependent on the response.



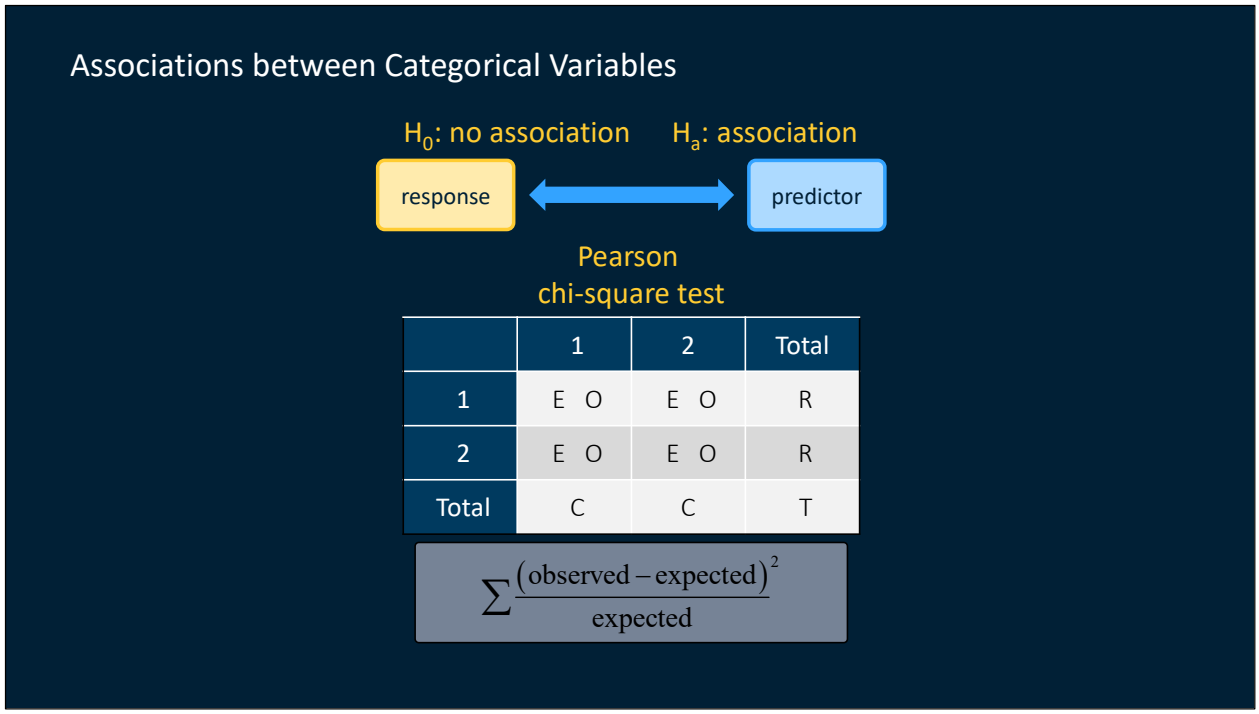
The chi-square measures the difference between the observed cell counts and the cell counts that are expected if there's no association between the variables and the null hypothesis is in fact true.

The expected counts for each cell is calculated by multiplying the row total (R) by the column total (C), and then dividing the result by the total sample size (T).

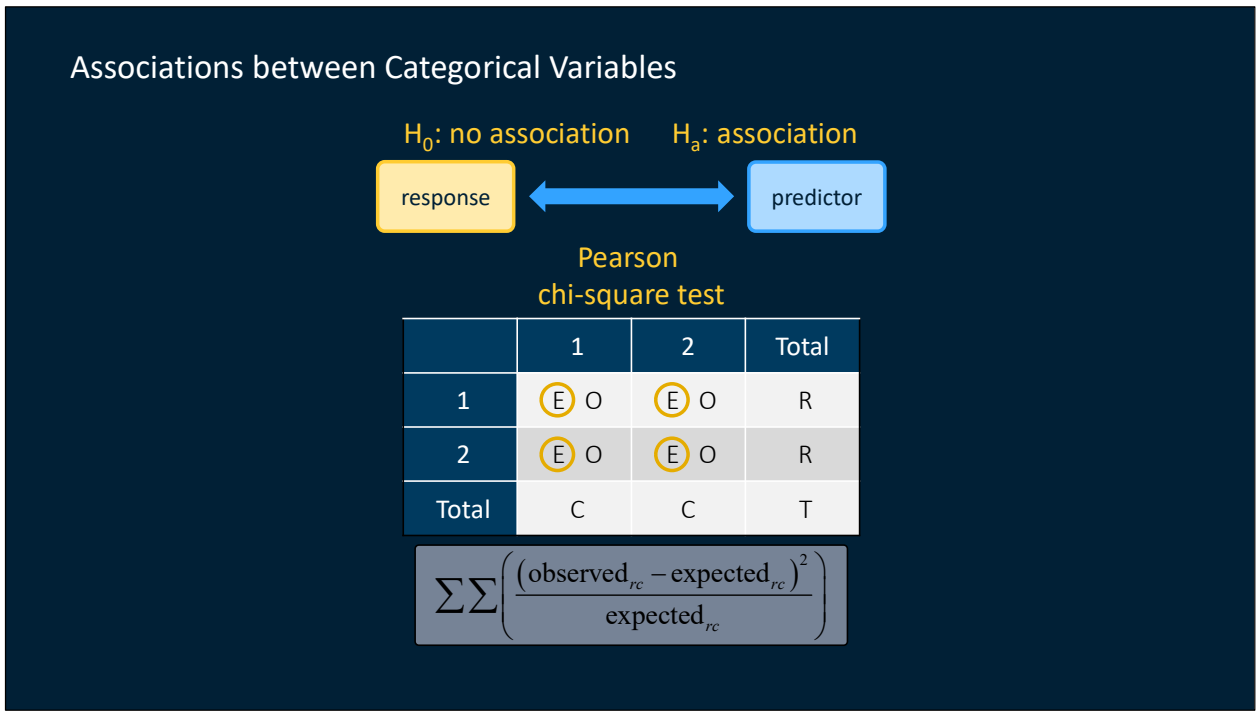


The larger the difference between the observed and expected cell counts, the more evidence of a statistically significant association between the variables.

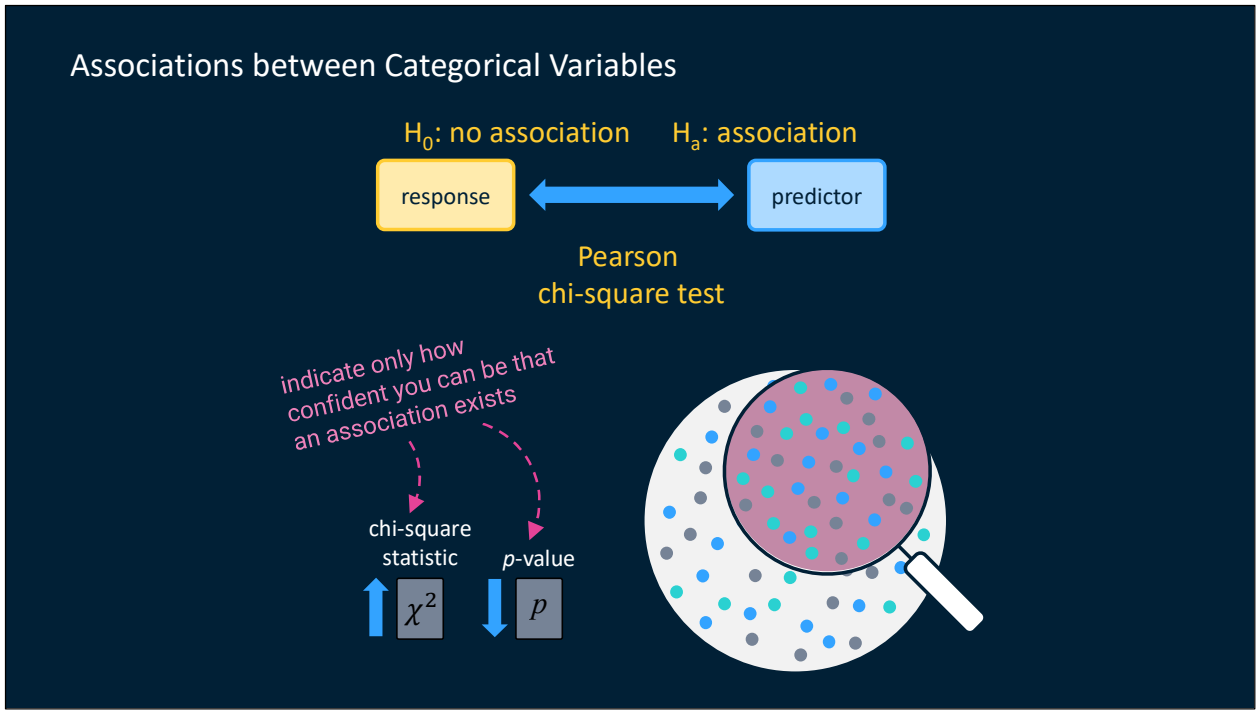
A significant chi-square statistic provides evidence to reject the null hypothesis and conclude that an association exists. To measure the magnitude of an association, we'll use measures of association such as Odds ratio.



To calculate the chi-square test statistic, you square the difference between the observed and expected counts for each cell, and divide by the expected cell count to get the cell chi-square values.



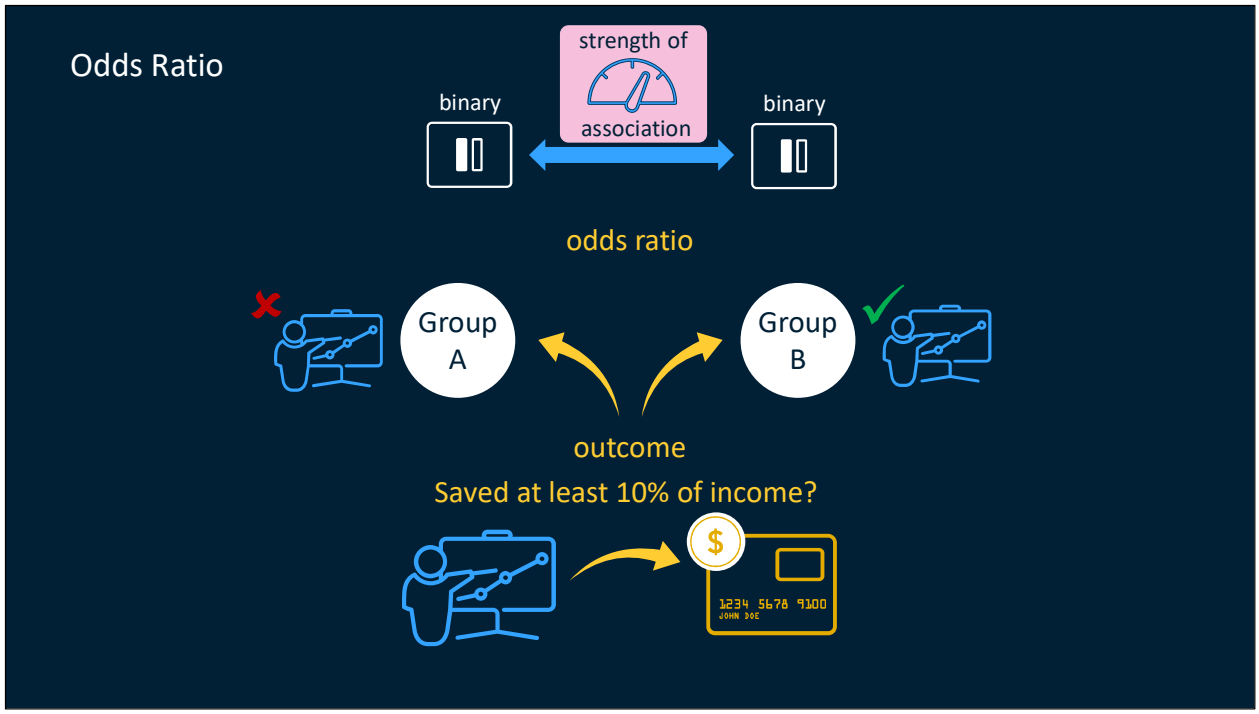
The overall chi-square, or the test statistic, is calculated by then adding the cell chi-square values over all cells.



Keep in mind that neither the chi-square statistic nor its *p*-value tells you the magnitude of an association. They indicate only how confident you can be that an association exists.

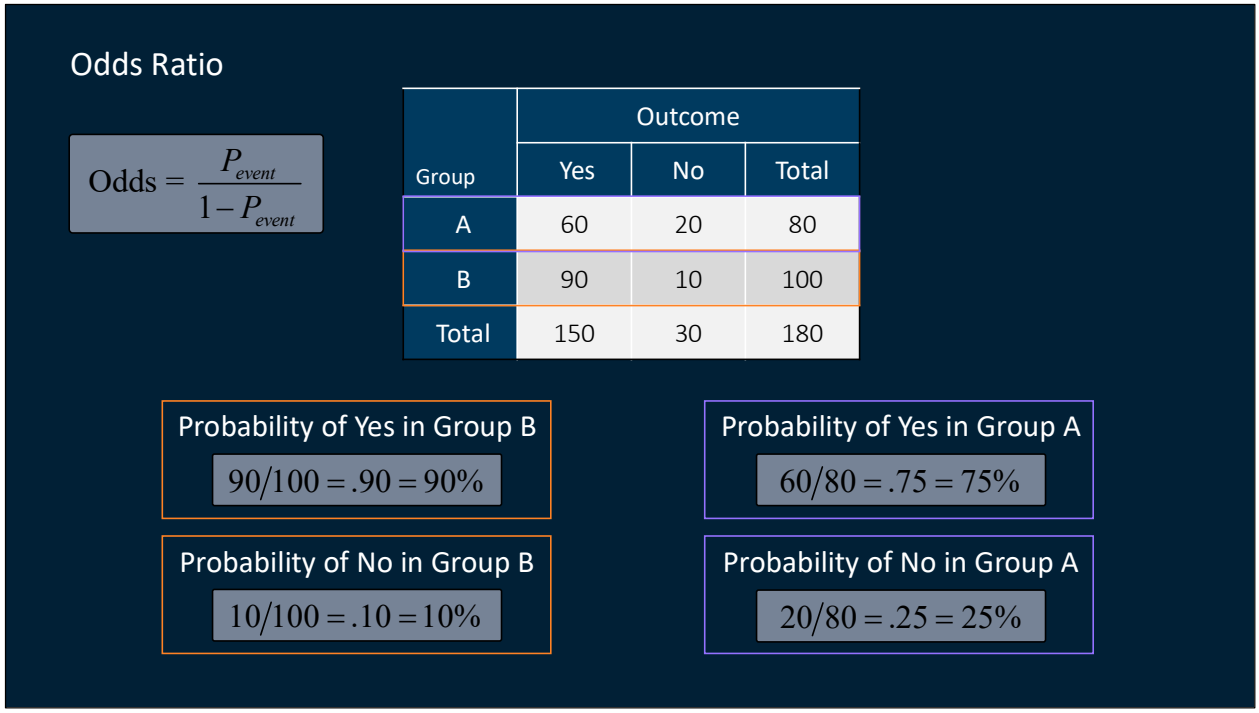
Chi-square statistics and their *p*-values depend on and reflect the sample size. A larger sample size yields a larger chi-square statistic and a smaller corresponding p-value, even though the association might not be strong.

For example, if you double the size of your sample by duplicating each observation, you double the value of the chi-square statistic, even though the strength of the association does not change.



To measure the strength of the association between a binary predictor variable and a binary response variable, you can use odds ratio. An odds ratio indicates how much more likely it is that a certain event, or outcome, occurs in one group relative to its occurrence in another group.

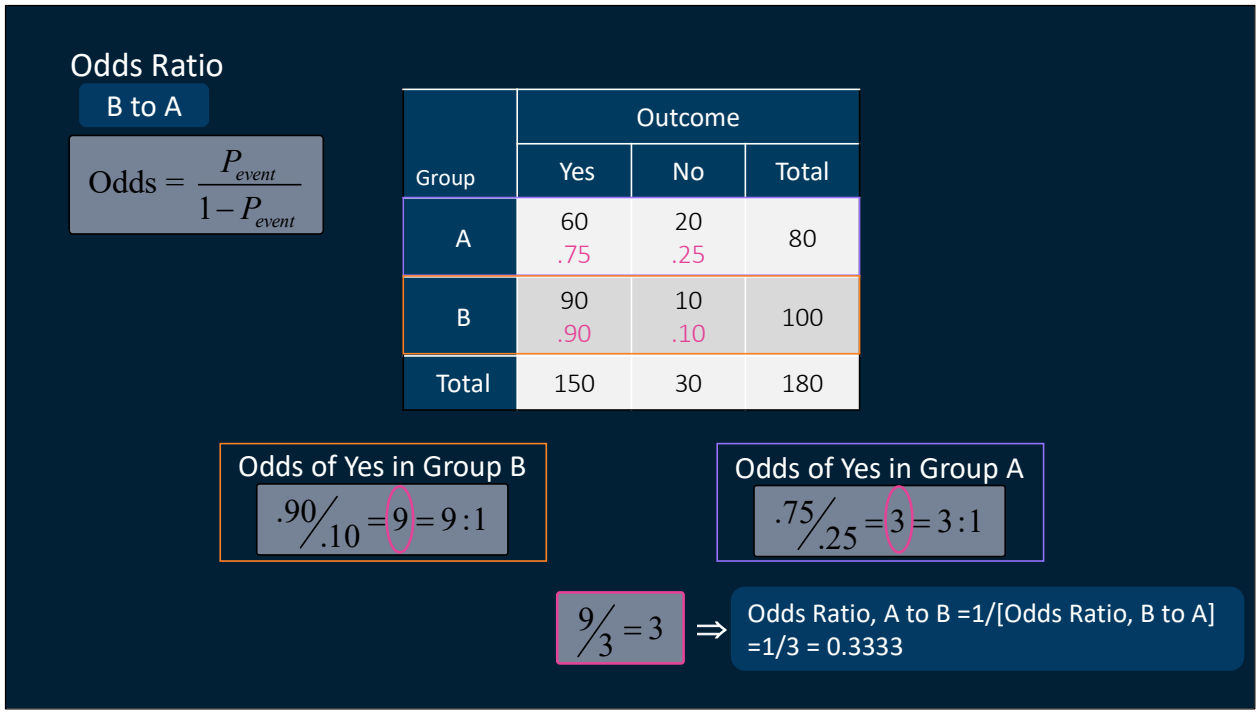
For example, suppose you want to see the effect that fiscal training has on spending habits. The people in Group B attended a seminar about saving money, and the people in Group A did not. The outcome variable indicates whether each person saved at least 10% of income in the year following the seminar. In this example, you want to know how much more likely it is that a Yes outcome occurs in Group B relative to its occurrence in Group A, with respect to the odds.



So, what are odds? Odds are not the same as probabilities. Instead, odds are calculated *from* probabilities. You divide the probability that the event occurs by the probability that the event does not occur.

To calculate the probability of a Yes outcome in Group B, you divide the number of Yes responses (90) by the total number of observations (100) to get 0.90, or 90%, the probability of having the Yes outcome in Group B. The probably of a No outcome in Group B is 10 divided by 100, which is 0.10, or 10%.

To calculate the probability of a Yes outcome in Group A, you divide 60 by 80, which is 0.75. The probability of a No outcome in Group A is 20 divided by 80, which is 0.25.

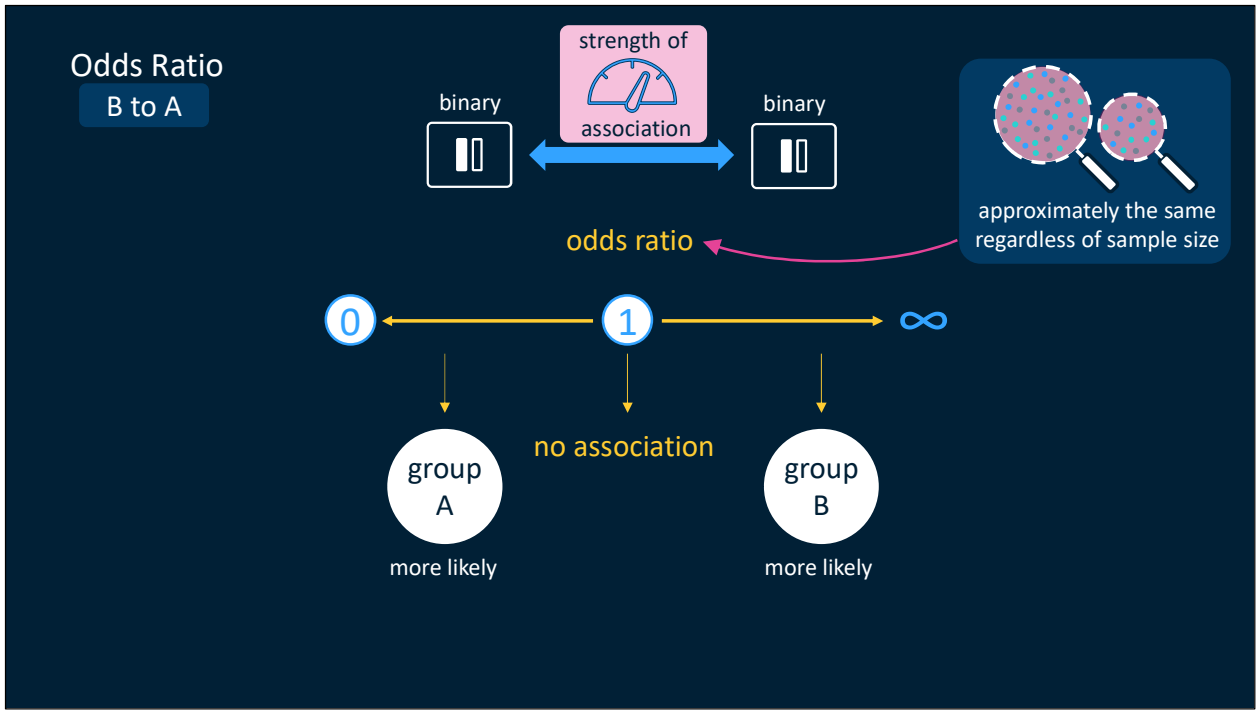


Now that we know the conditional probabilities of our response given the predictor, let's calculate the odds. The odds of the outcome occurring in Group B are the probability of a Yes outcome, 0.90, divided by the probability of a No outcome, 0.10. The odds are 9, or 9:1, which means that we expect nine occurrences to one non-occurrence in Group B.

To calculate the odds of the outcome occurring in Group A, you divide the probability of a Yes outcome, 0.75, by the probability of a No outcome, 0.25, which is 3, or 3:1. This means that you expect three times as many occurrences as non-occurrences in Group A.

Now that you know the odds of the outcome in both groups, you can compare the odds by calculating an odds ratio. You divide the odds of an outcome in Group B, 9, by the odds in Group A, 3, with a result of 3. An odds ratio of 3 means that the odds of getting the outcome in Group B are three times those of getting the outcome in Group A. So, in this example, the odds of saving at least 10% of income are three times higher for people who received training.

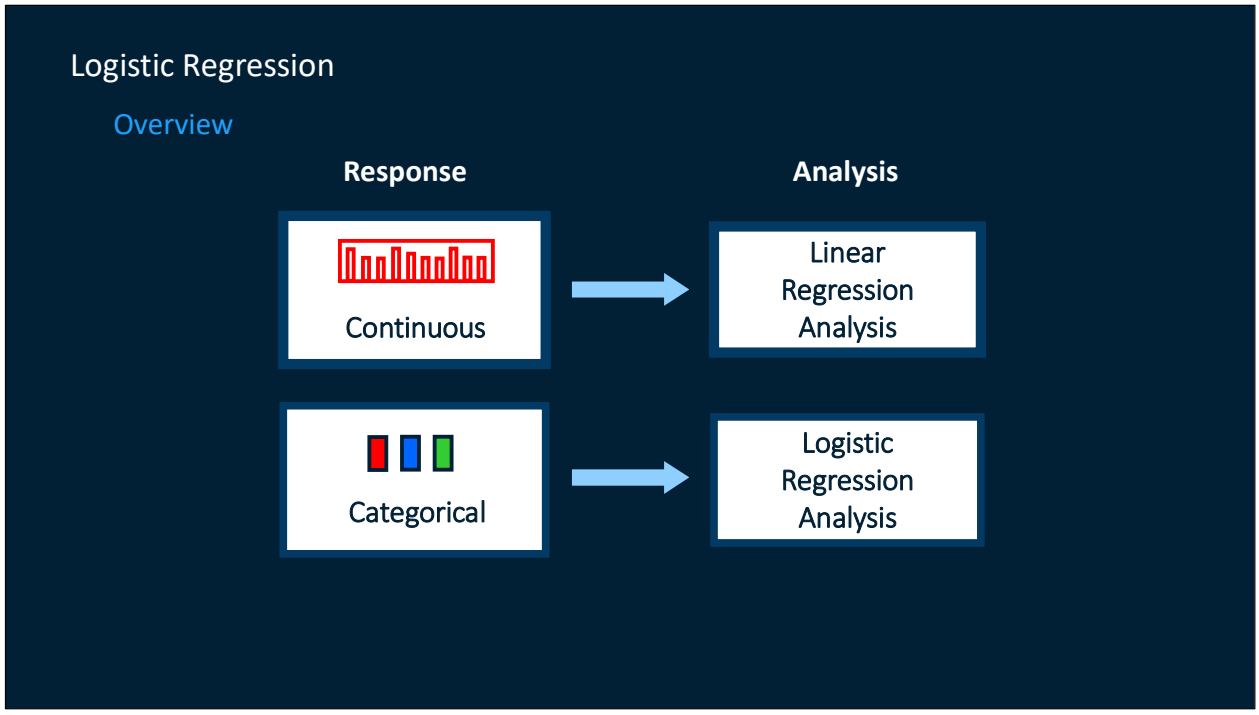
If you were interested in the odds ratio of group A to group B, you would simply take the multiplicative inverse (or reciprocal) of 3 to arrive at 0.3333, which indicates that the odds of getting the outcome in group A are one-third those in group B.



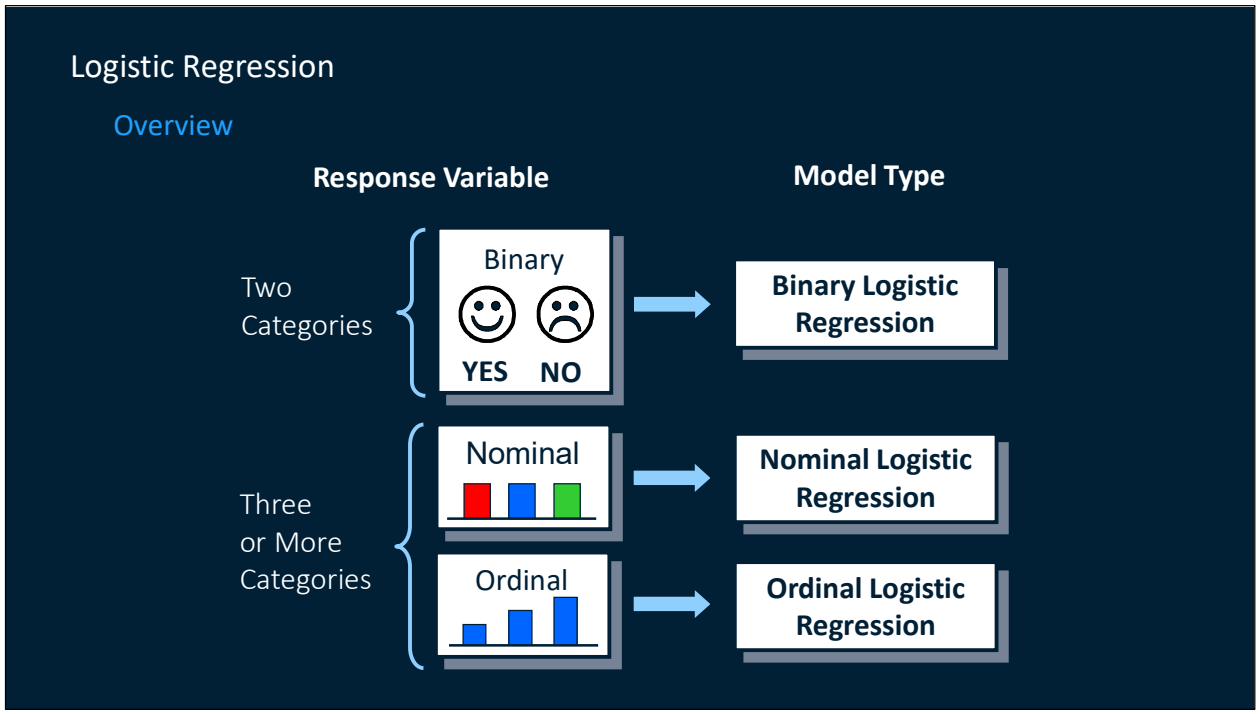
The odds ratio shows the strength of the association between the predictor variable and the outcome variable, and the value can range from 0 to infinity.

When the odds ratio is 1, there's no association between the predictor variable and the outcome variable. If the odds ratio is greater than 1, the group in the numerator (here Group B) is more likely to have the outcome. If the odds ratio is less than 1, the group in the denominator (here Group A) is more likely to have the outcome. The odds ratio is approximately the same regardless of sample size.

4.3 Logistic Regression Model



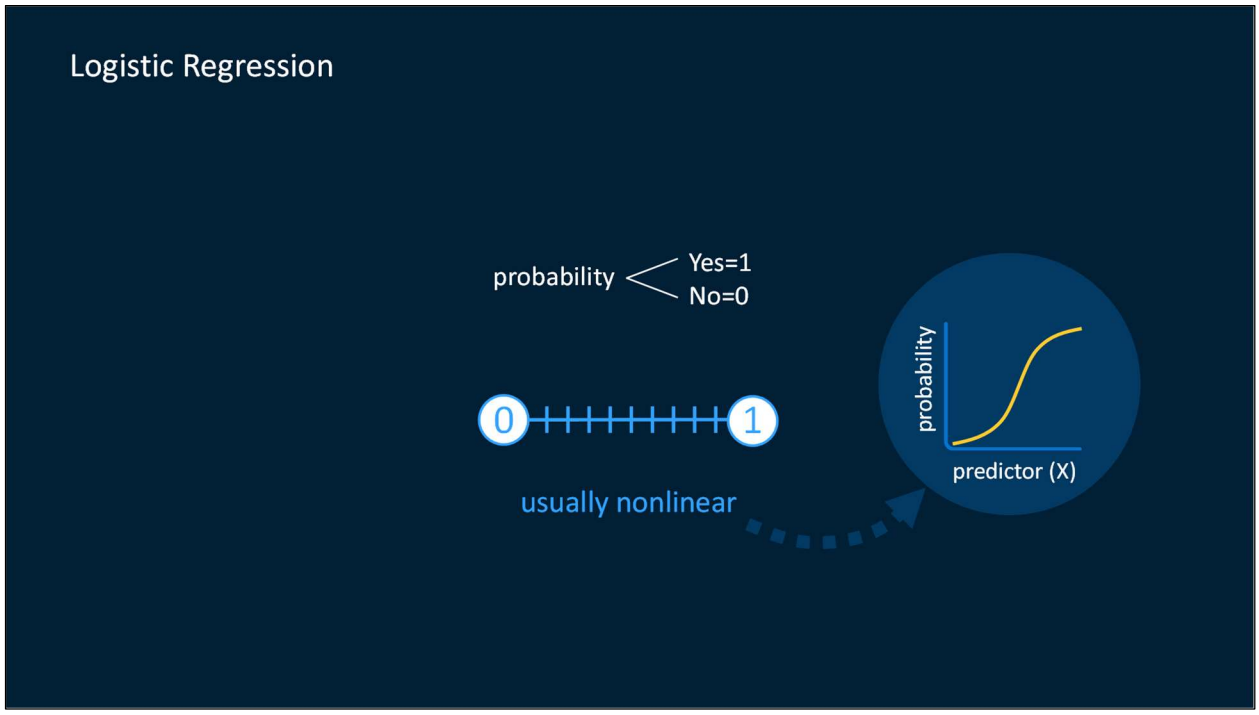
Regression analysis enables you to characterize the relationship between a response variable and one or more predictor variables. In linear regression, the response variable is continuous. In *logistic regression*, the response variable is categorical.



If the response variable is dichotomous (meaning it has two categories), the appropriate logistic regression model is binary logistic regression. This is the most common logistic regression model in data mining and machine learning applications, so it's the primary focus in this course.

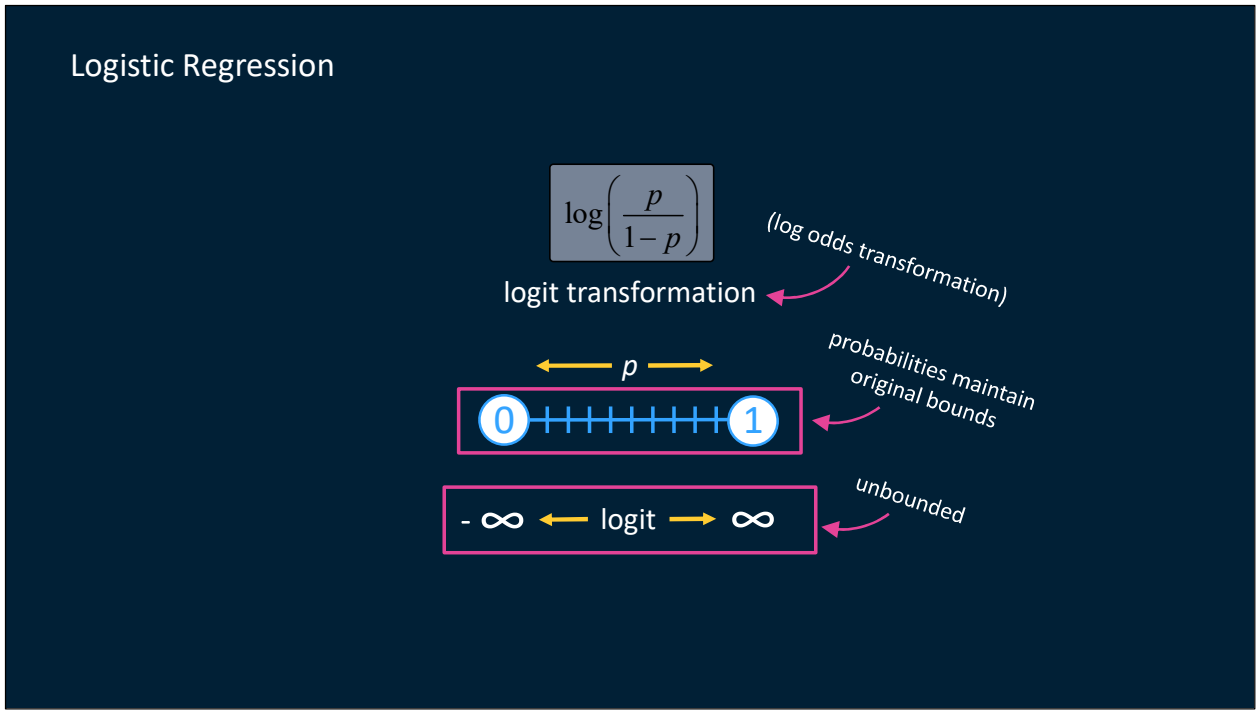
If you have more than two categories (or levels) within the response variable, there are two possible logistic regression models:

- If the response variable is nominal, you fit a nominal logistic regression model.
- If the response variable is ordinal, you fit an ordinal logistic regression model.



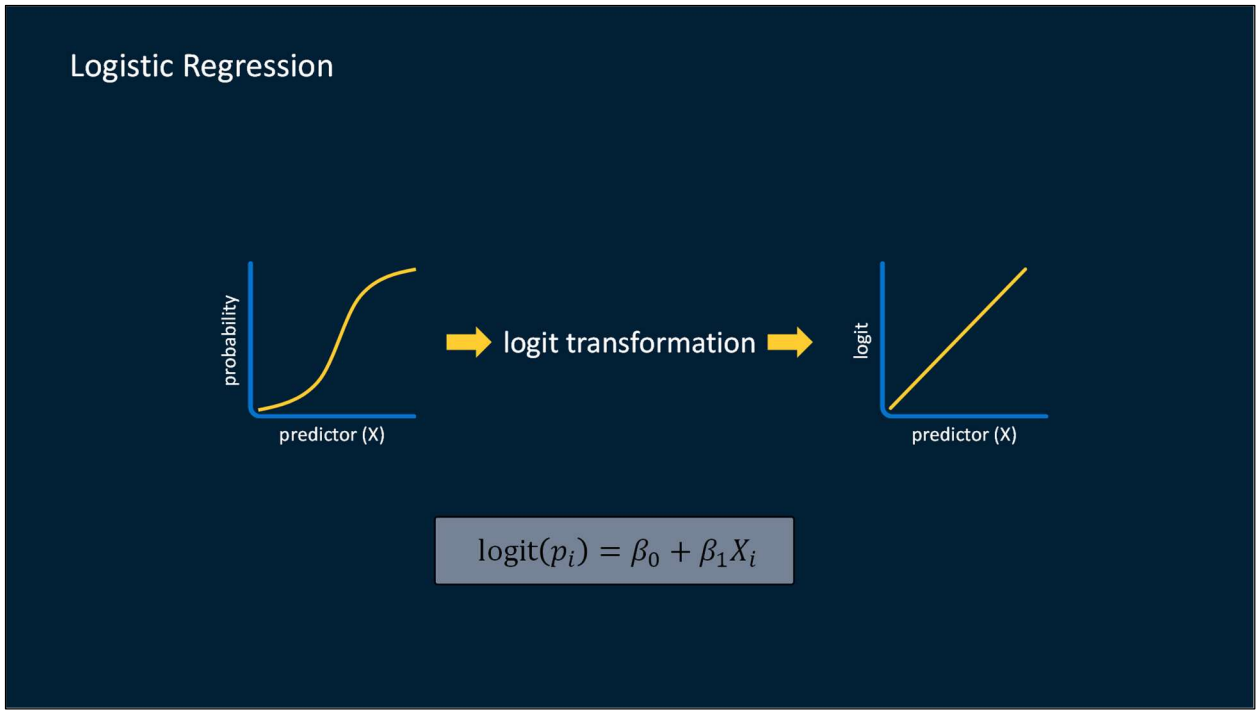
In logistic regression, we want to model the probability of both levels of the response. Although probabilities are continuous, they are bounded between 0 and 1. Also, the relationship between the probability of the outcome and a predictor variable is usually nonlinear rather than linear.

In fact, the relationship often resembles an S-shaped, or sigmoidal, curve.



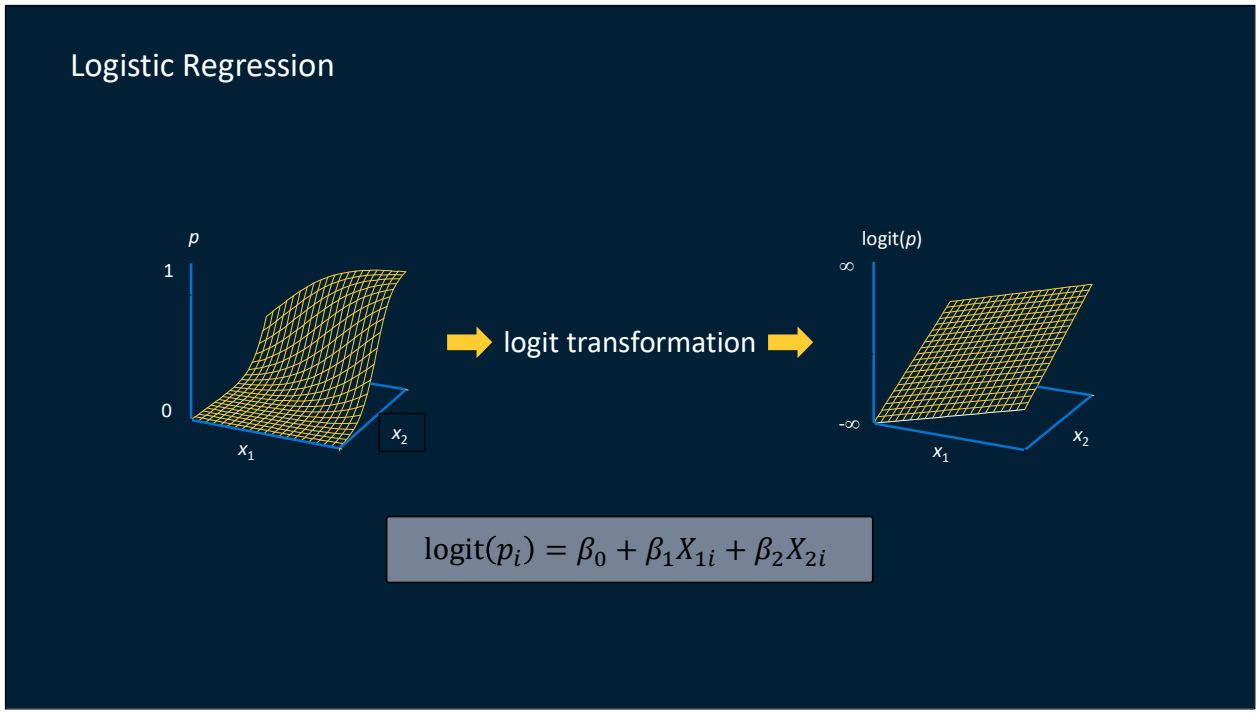
A logistic regression model applies a logit transformation, or simply the log odds transformation, to the probabilities. Here, log means 'natural' log, rather than log with a base 10. The logit effectively avoids the boundary problem for probabilities.

As *p* approaches its minimum value of 0, the logit approaches negative infinity, and as *p* approaches its maximum value of 1, the logit approaches positive infinity. So, the logit is unbounded, just like in linear regression, but the probabilities maintain the original bounds of 0 to 1.



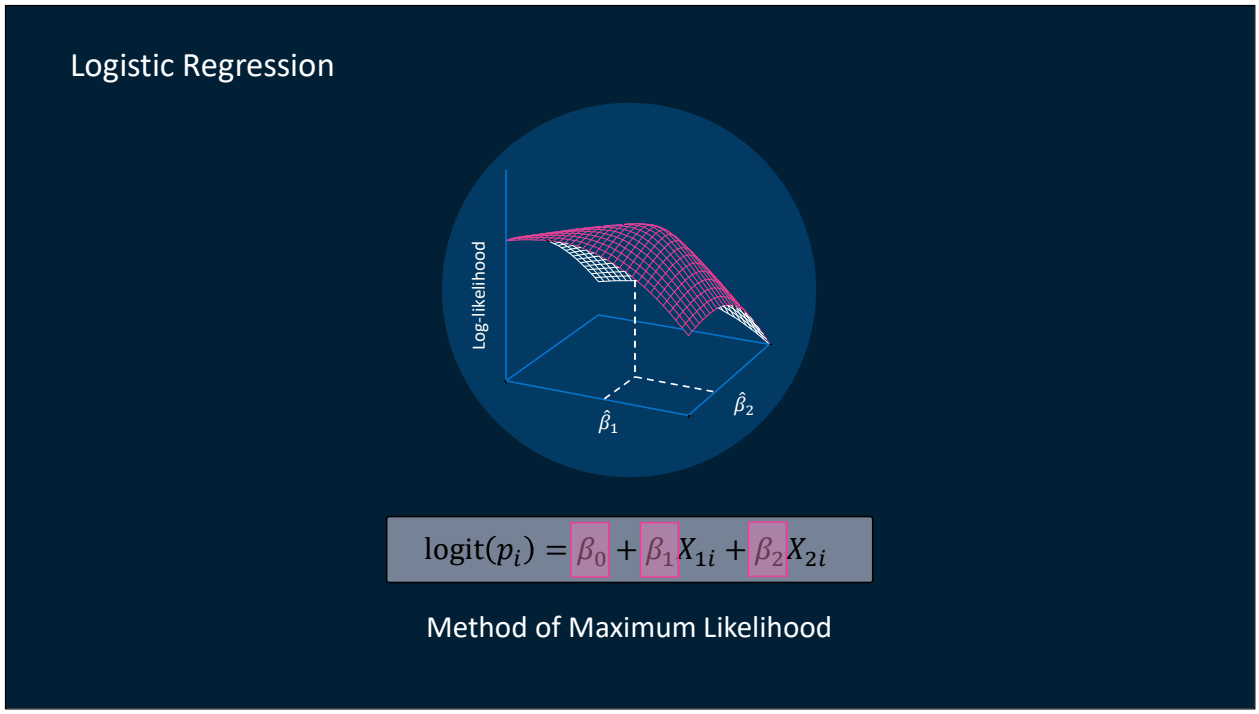
In addition, the logit transformation enables us to move from modeling the probability with a sigmoidal nonlinear curve to modeling the logit with a linear function of the predictors. And modeling the logit allows you to indirectly model the probability of the response. Whatever the predicted value of the logit is, we can simply back-transform to the probability scale to get a value between 0 and 1.

The logistic model now looks quite familiar, compared to linear regression. In a two-dimensional space, with a binary target and one predictor (X), the logit is equal to a linear function of your parameters and predictor. Again, β_0 is the intercept of the regression equation, and β_1 is the slope of the model predictor. Unlike linear regression, we no longer assume normally distributed errors in logistic regression. Here the error is a function of p, the probability of the event.

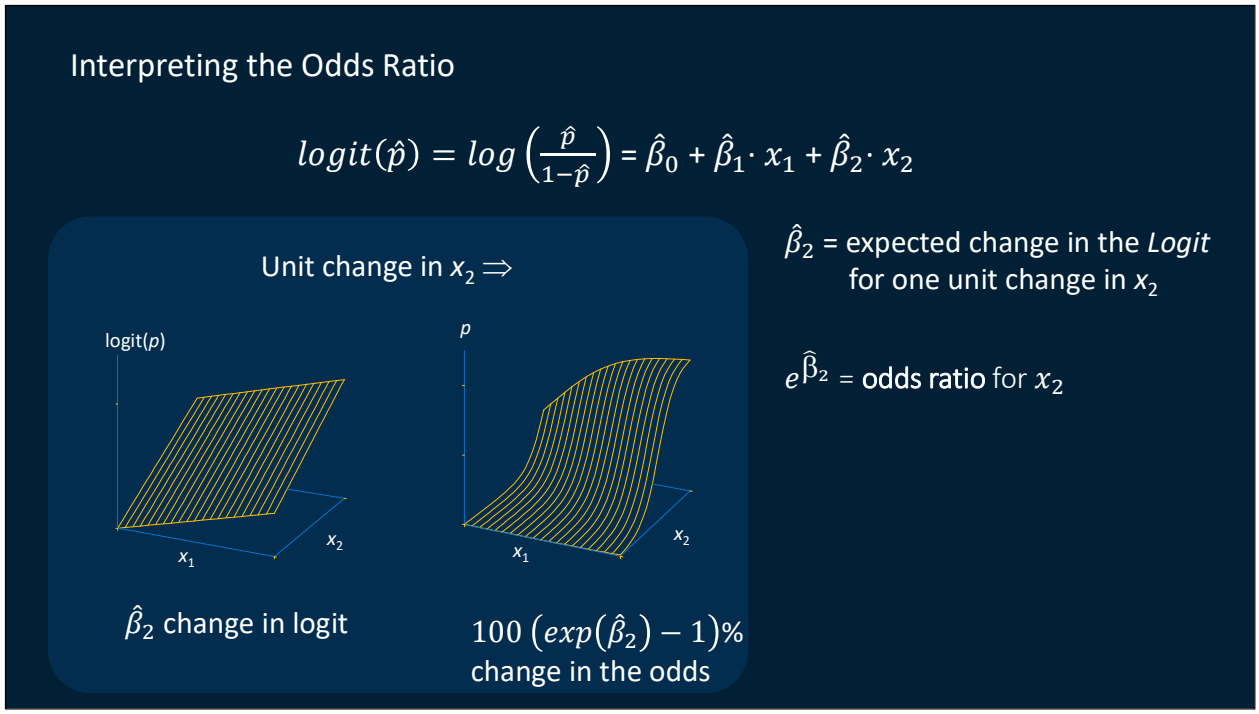


In a three-dimensional space, with a binary target and two predictors (X_1 and X_2), the logit is equal to a linear function of your parameters and predictors. Again, β_0 is the intercept of the regression equation, and the other betas are the slopes of the model predictors.

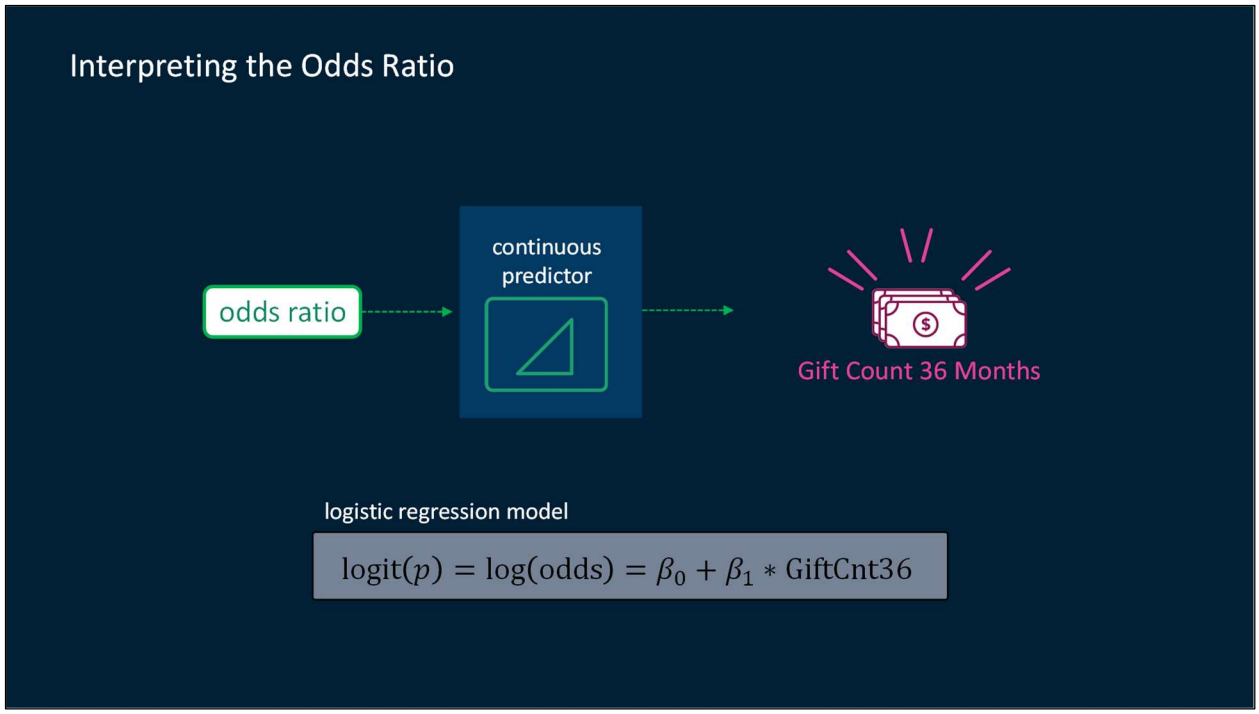
The graph of a linear combination on the logit scale is a (hyper)plane. Different parameter values give different surfaces with different slopes and different orientations. On the probability scale, it becomes a sigmoidal surface. The nonlinearity of the model is used to deal with the constrained scale of the target.



To fit the model, logistic regression requires a more computationally complex estimation method, named the method of maximum likelihood, to estimate the parameters. This method finds the values of the parameters that make the observed data most likely. This is accomplished by maximizing the likelihood function that expresses the probability of the observed data as a function of the unknown parameters.

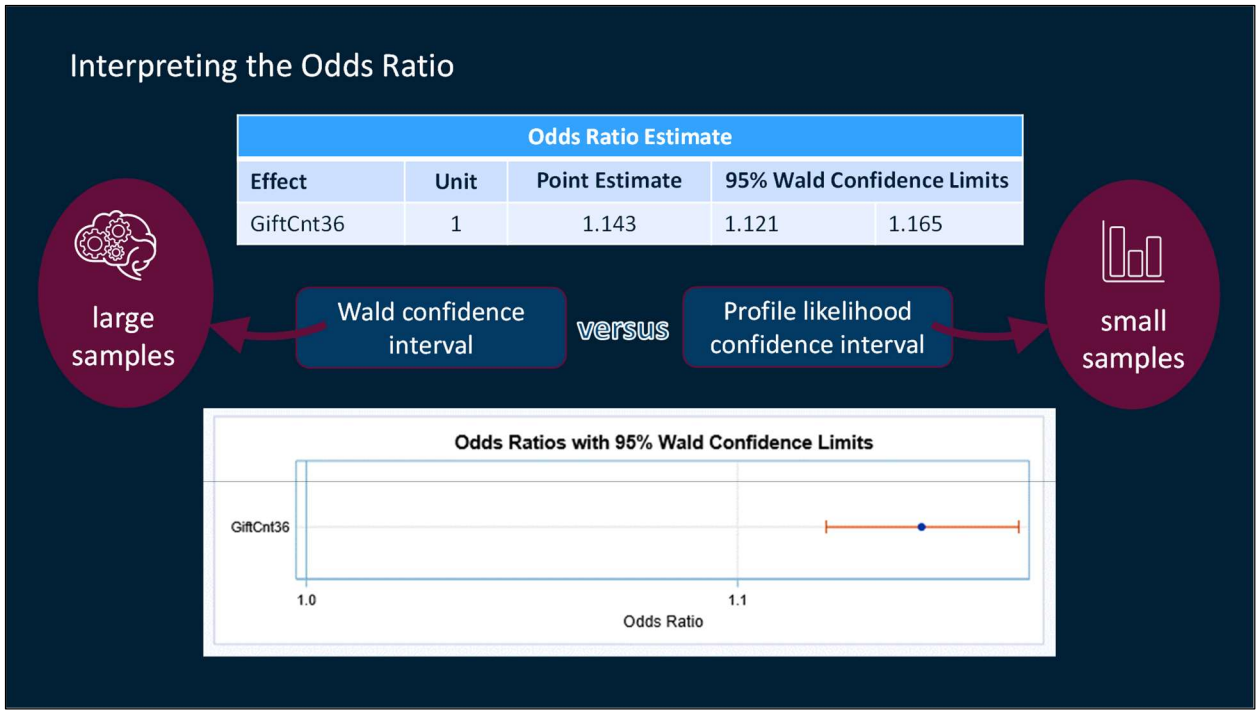


A linear-additive model is particularly easy to interpret since each input variable affects the logit linearly. The coefficients are the slopes. Exponentiating each parameter estimate gives the odds ratios, which compares the odds of the event in one group to the odds of the event in another group.



To help interpret the odds ratio, let's see how to calculate the odds and the odds ratio from the logistic regression model. For a continuous predictor variable such as **GiftCnt36**, the odds ratio measures the increase or decrease in odds associated with a one-unit difference of the predictor variable.

Remember, the logit is the natural log of the odds. Because you can calculate an estimated logit from the logistic model, the odds can be calculated by simply exponentiating that value. An odds ratio for a one-unit difference is then the ratio of the exponentiated predicted logits that are one unit apart.

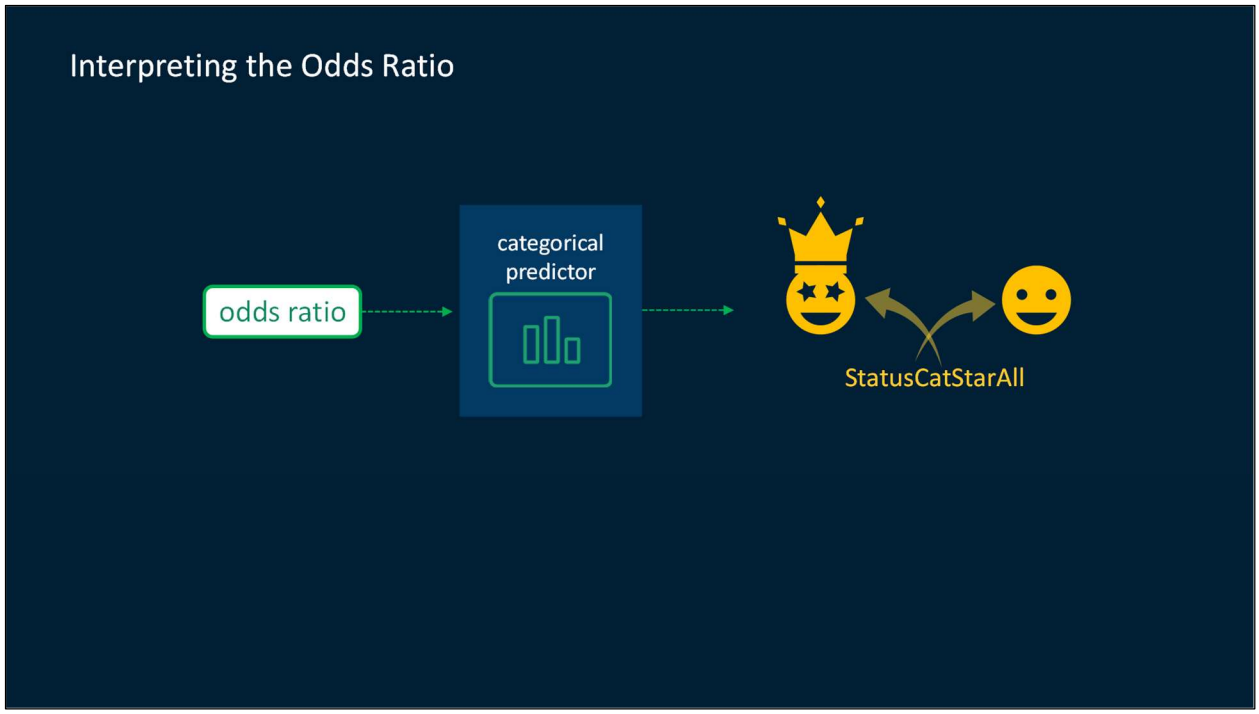


For **GiftCnt36**, the odds ratio estimate equals 1.143. This means that for each additional donation in the past 36 months, the odds of donation during the 97NK campaign change by a factor of 1.143, a 14.3% increase.

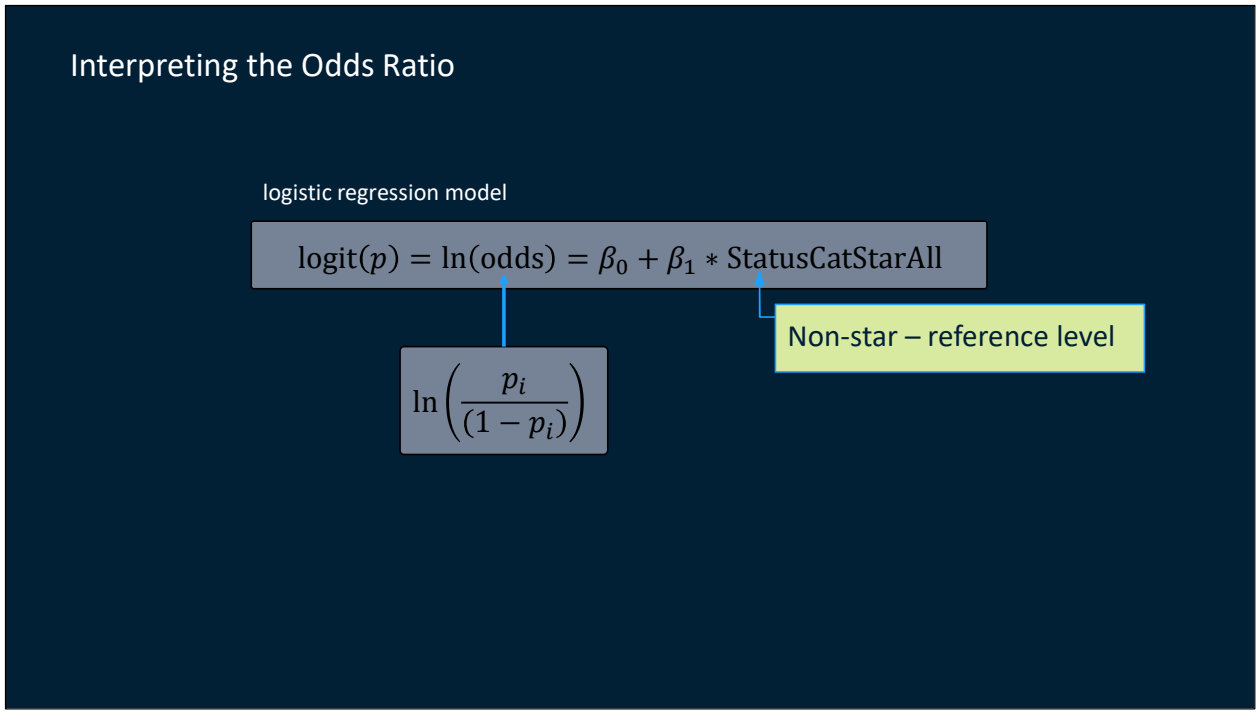
Because the 95% confidence interval, 1.121 to 1.165, does not include 1.000, the odds ratio is significant at the 0.05 alpha level, and therefore, the predictor **GiftCnt36** is significantly different from 0.

Note that the Wald-based confidence intervals are particularly suitable in the machine learning world, but different from profile likelihood confidence intervals you had seen earlier. This difference is because the Wald confidence intervals use a normal error approximation, whereas the profile likelihood confidence intervals are based on the value of the log-likelihood. These likelihood-ratio confidence intervals require a much greater number of computations, but are generally preferred to the Wald confidence intervals, especially for sample sizes less than 50.

The Odds Ratio plot displays the results of the Odds Ratio table graphically. This plot is obtained by applying the parameter estimates from the logistic model to values of the predictors, and then converting the predictions to the probability scale. A reference line shows the null hypothesis, an odds ratio equal to 1. When the confidence interval crosses the reference line, the effect of the variable is not significant.



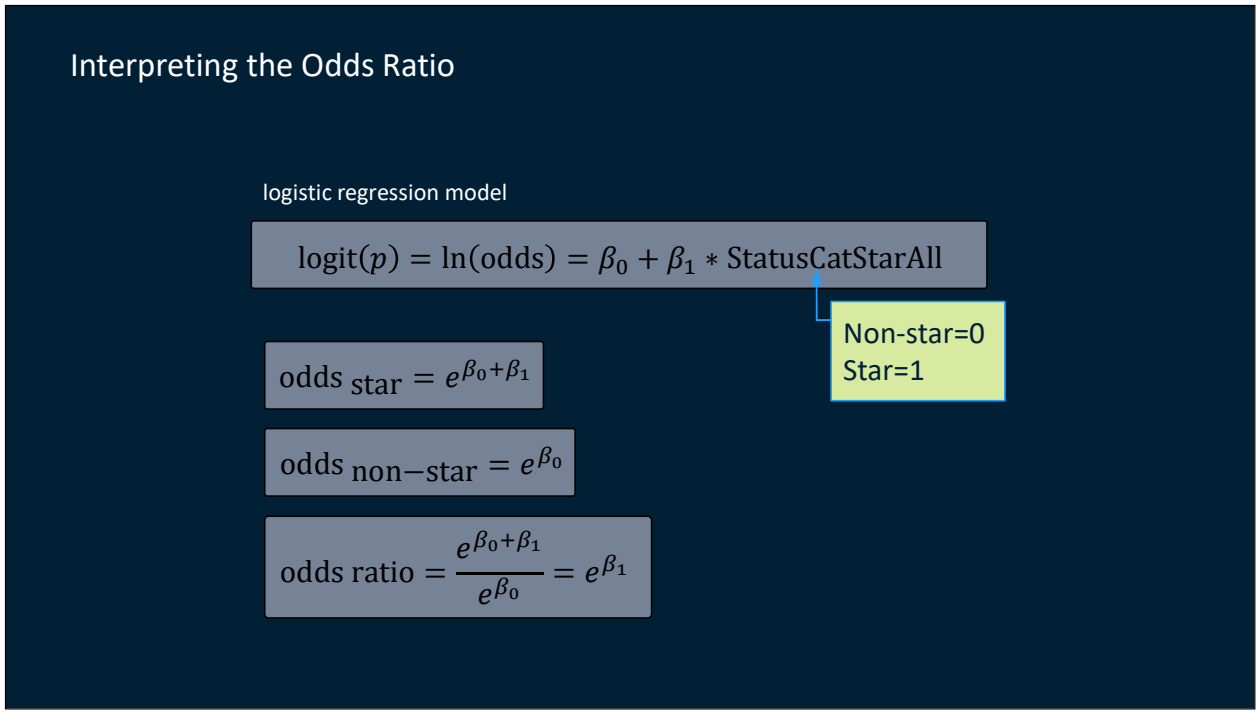
Calculating and interpreting odds ratios for categorical variables is similar to that of continuous variables. Consider a categorical predictor variable such as **StatusCatStarAll** that has two levels, star donors and non-star donors.



Imagine now that we fit a logistic regression model with the predictor **StatusCatStarAll** instead of **GiftCnt36**.

The logit of p is also equal to the linear predictor for our model: $\beta_0+\beta_1$*StatusCatStarAll.

In this case, we use the level non-star to represent the reference level.

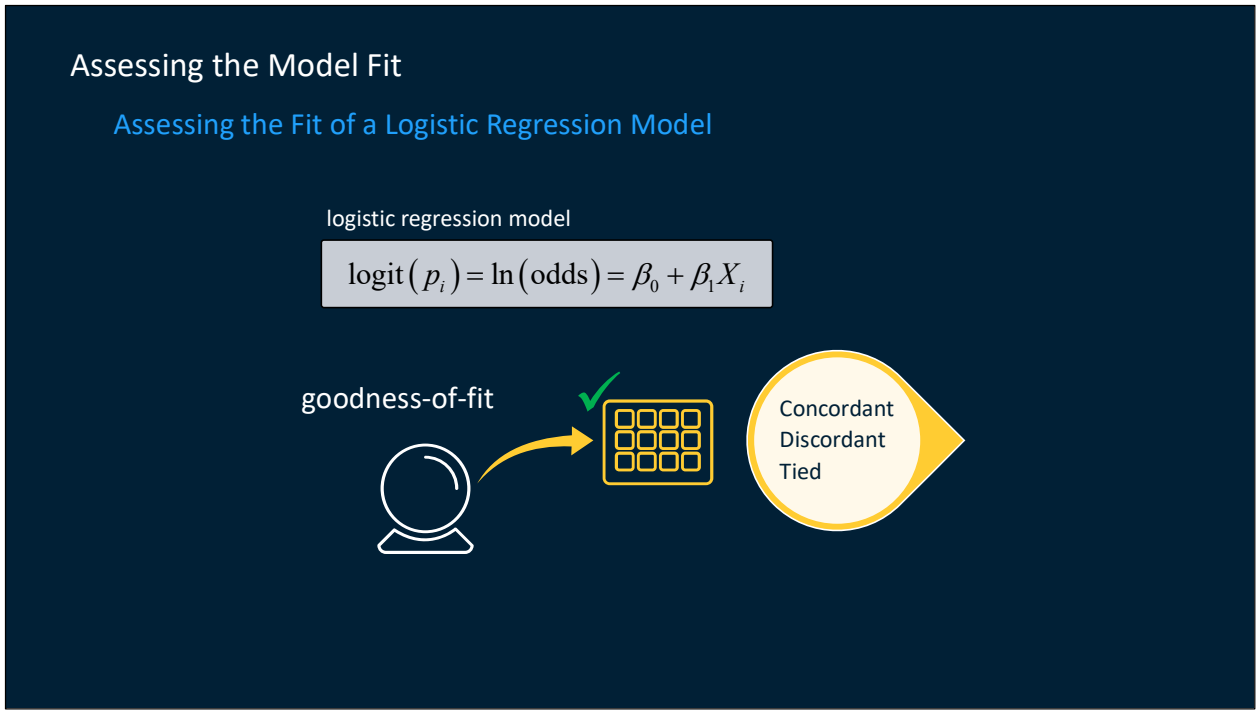


So, individuals who belong to the non-star category are coded as 0 and those who belong to star category are coded as 1.

To obtain the odds for a star category, we exponentiate the linear predictor for the level. First, we substitute 1 for StatusCatStarAll to get $\beta_0+\beta_1$ as the linear predictor. Then, we add the parameter estimates that we got for β_0 and β_1 and exponentiate the sum.

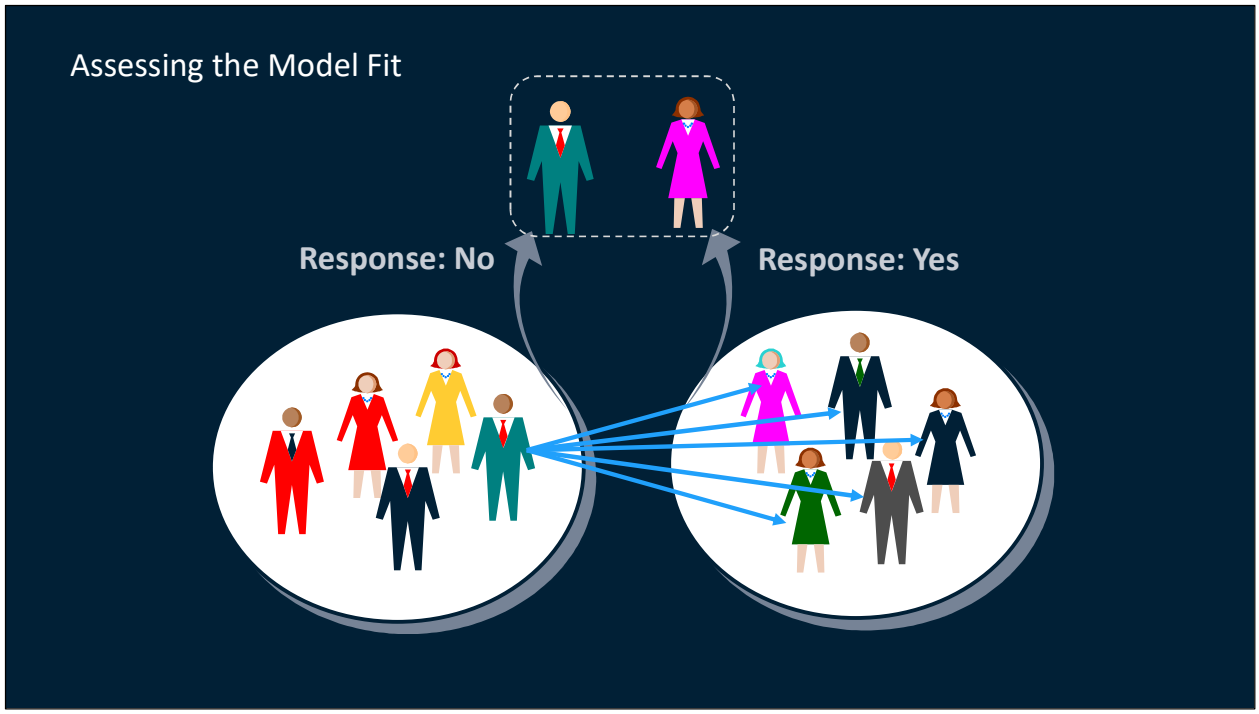
To obtain the odds for a non-star category, we follow the same process. First, we substitute 0 for StatusCatStarAll to get β_0 as the linear predictor. Then we take the parameter estimate that we got for β_0 and exponentiate it.

The odds ratio is then the odds for the star category divided by the odds for a non-star category. Mathematically, this is equivalent to $e^{\beta 1}$, so we can divide the two values we just calculated, or we can simply take the parameter estimate that we got for β_1 and exponentiate it.



If a logistic regression model predicts its own data accurately, then we say that the model fits the data well.

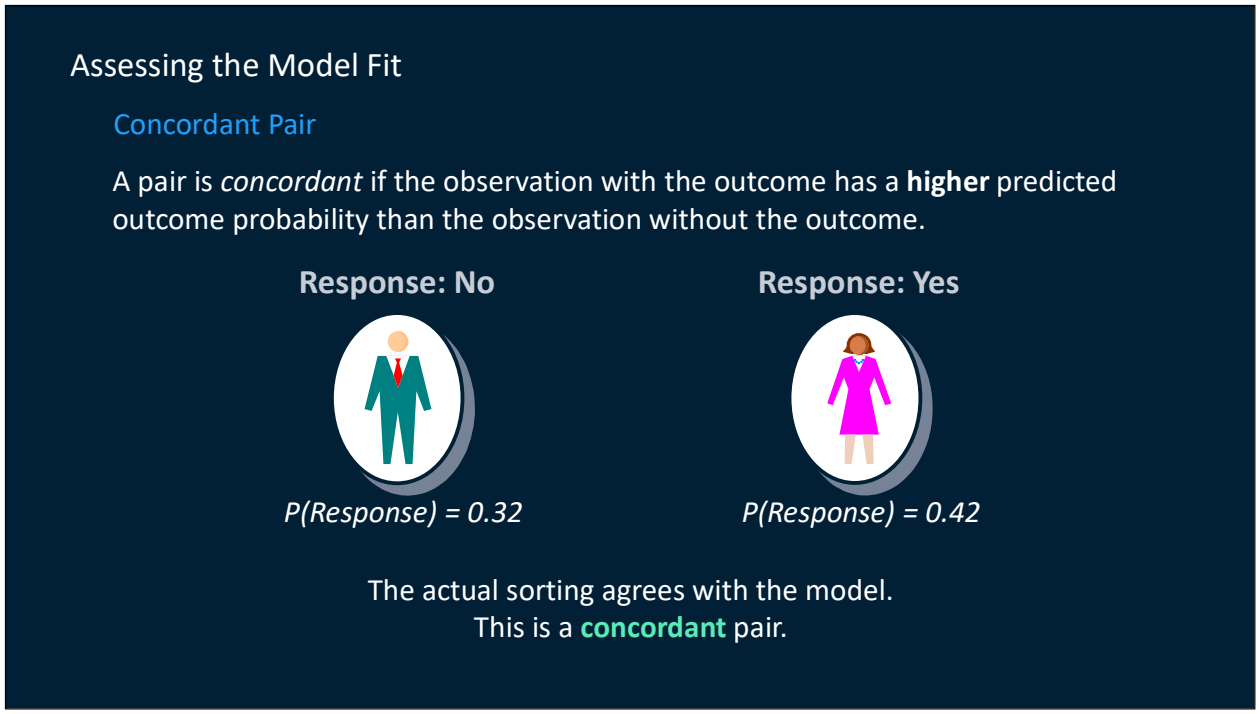
Counting concordant, discordant, and tied pairs is a way to assess how well the model predicts its own data and therefore how well the model fits. In general, you want a high percentage of concordant pairs and low percentages of discordant and tied pairs.



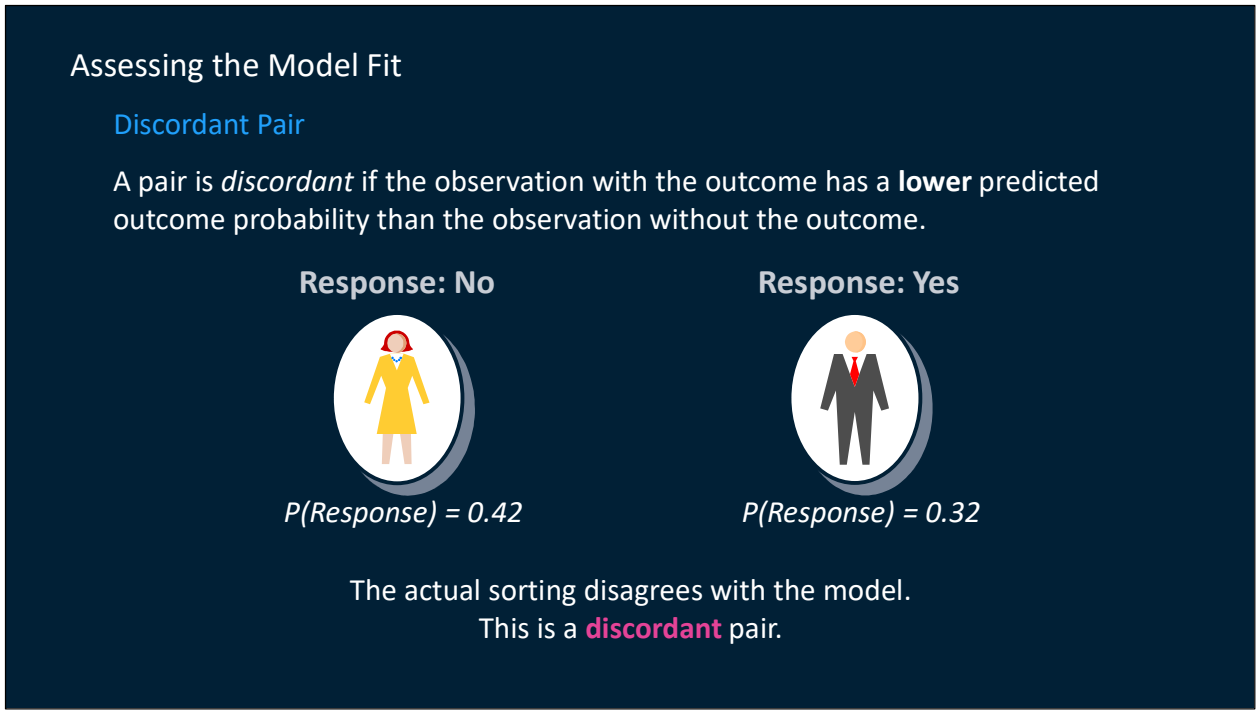
To find concordant, discordant, and tied pairs, we compare everyone who had the outcome of interest against everyone who did not.

SAS creates two groups of observations, one for each value of **Response**. One group contains all the observations in which the value of **Response** is 0, meaning the non-responders. The other group contains all the observations in which the value of **Response** is 1, the responders.

Each member in each group is paired to every member in the other group. One such pair is shown here. SAS then selects every pair and then determines whether each pair is concordant, discordant, or tied. Let's look at each type of pair.



Suppose that a pair consists of a woman who responded to the campaign and a man who did not. The man who did not respond to the campaign has a lower predicted probability of response, .32, than the woman who responded to the campaign, 0.42. A pair is concordant when the observation with the event, in this case, response=Yes, has a higher predicted probability of having the event than the observation without the event. That is, the model assigns a higher probability of being a responder than to the non-responder. When the model sorts the pair correctly, as in this case, the pair is concordant.



Let's look at another pair. This pair compares a man who responded to the campaign and a woman who did not. From our model, we know that the non-responder woman have a higher predicted probability, 0.42, of being a responder, than the man who actually responded to the campaign, 0.32. However, our model did not sort this pair correctly. A pair is discordant if the observation with the desired outcome has a *lower* predicted probability than the observation without the outcome.

Assessing the Model Fit

Tied Pair

A pair is *tied* if it is neither concordant nor discordant. The predicted outcome probabilities are the **same**.

Response: No

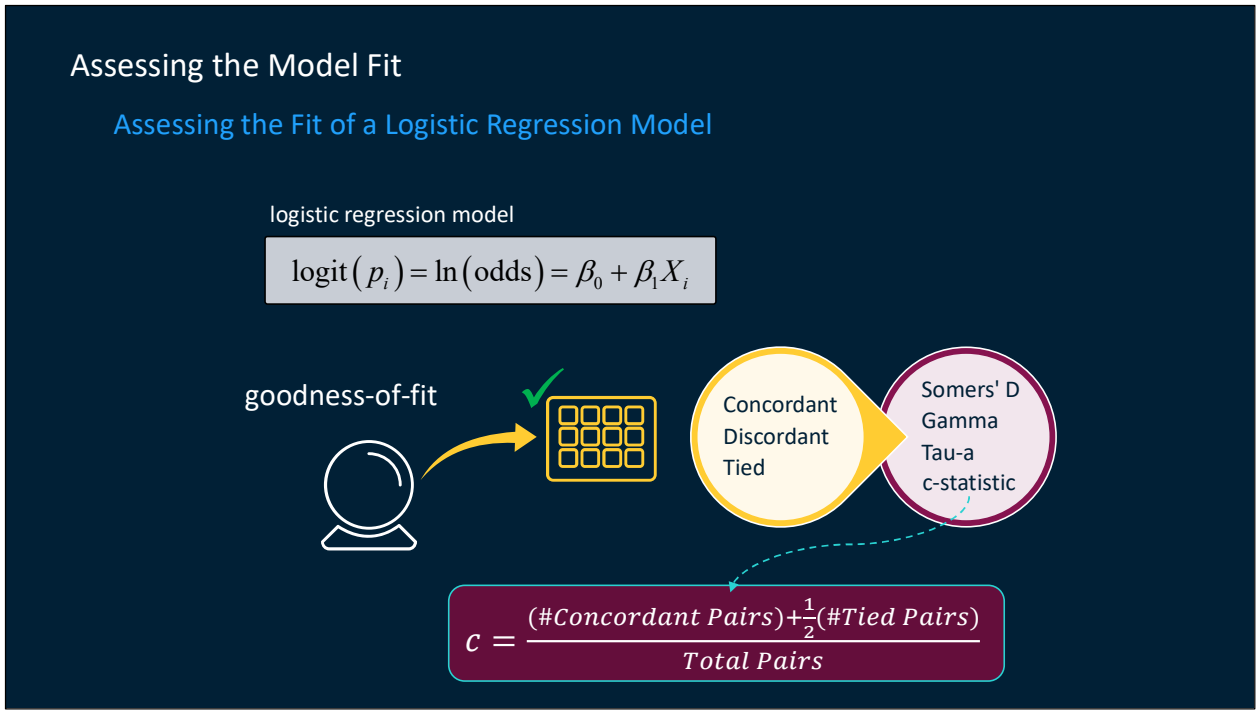
P(Response) = 0.42

Response: Yes

P(Response) = 0.42

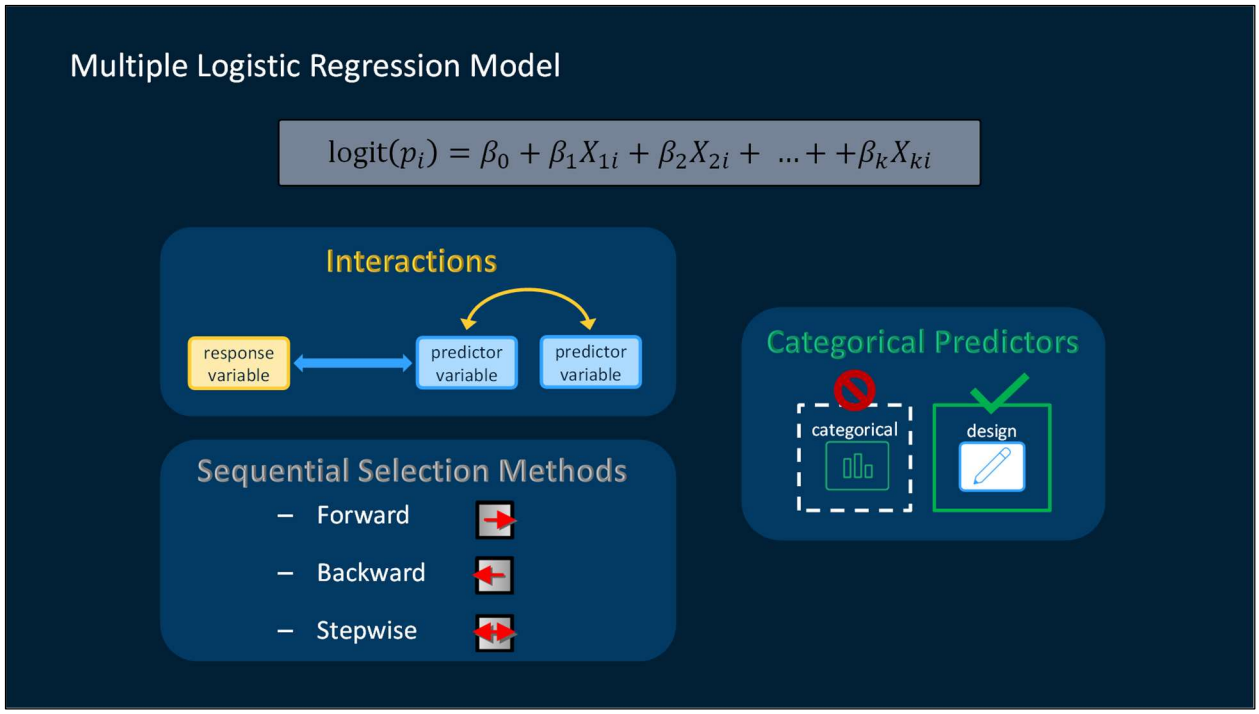
The model cannot distinguish between the two.
This is a **tied** pair.

The last pair compares two women, one who responded to the campaign and the other who did not. According to our model, both have a predicted probability of response as 0.42, so our model cannot distinguish between them. A pair is tied if it is neither concordant nor discordant, that is, the probabilities are the same.



Several rank correlation indices are also computed from the numbers of concordant, discordant, and tied pairs of observations: Somers' D, Gamma, Tau-a, and c. In general, a model with higher values for these indices has better predictive ability than a model with lower values.

The c value is the most commonly used. The c, concordance statistic, estimates the probability of an observation with the event having a higher predicted probability than an observation without the event. The c value is calculated as the number of concordant outcomes plus one-half times the number of ties divided by the total number of pairs. The range of possible values is 0.5 to 1.0, where 1.0 is perfect prediction. It can be shown that the area under the ROC curve for a model equals the c-statistic. Thus, the c-statistic also called the ROC index.

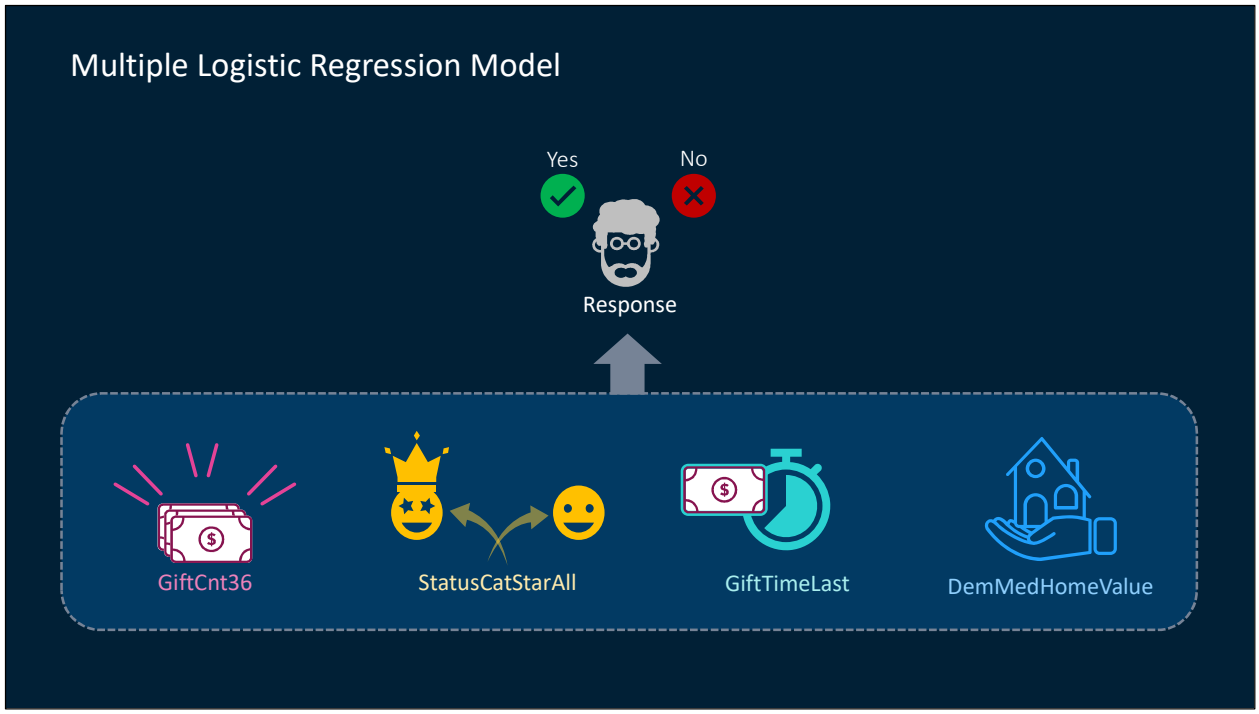


Just as in linear regression, if you have many predictors in logistic regression, it is called multiple logistic regression model. A multiple logistic regression model with *k* predictors is shown here. The number of parameters in the logistic model consider the intercept and the number of predictors.

Just as in linear regression, if you suspect that there are interactions between predictors, you can fit a more complex logistic regression model by including interaction effects. Remember, an interaction is present when the effect of one variable on the outcome depends on, or changes, due to another variable.

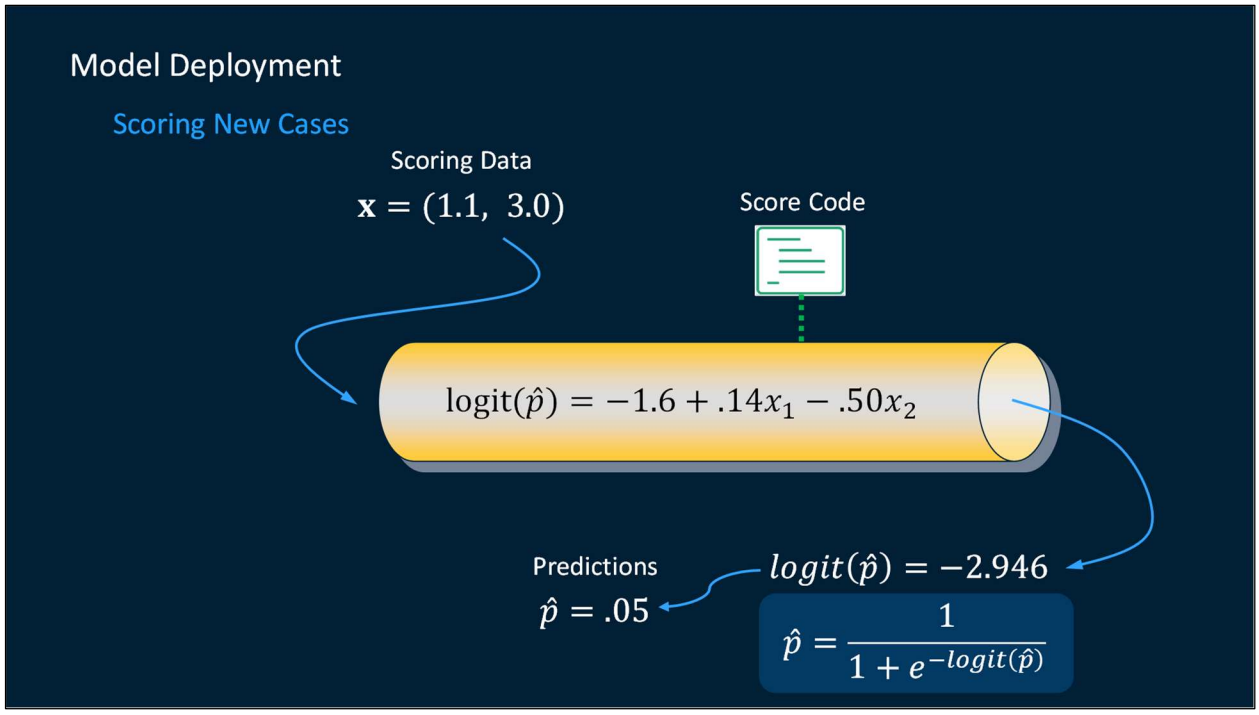
Just as in linear regression, a set of design variables, also known as dummy variables, is created for categorical predictors that represent the information in each classification variable. There are various methods of parameterizing the classification variables.

Just as in linear regression, you have similar model selection options in logistic regression that commonly includes forward, backward, and stepwise methods.



Now, let's fit a logistic regression model using some continuous and categorical predictors. You already saw that Response and StatusCatStarAll have a significant association. In addition to this, you use three more continuous variables: Gift count 36 months, Gift time last, and Dem Median home value to fit a multiple logistic regression model. However, you can always use the available model selection methods to select inputs for model building.

4.4 Model Deployment



The overriding purpose of predictive modeling is to score new cases.

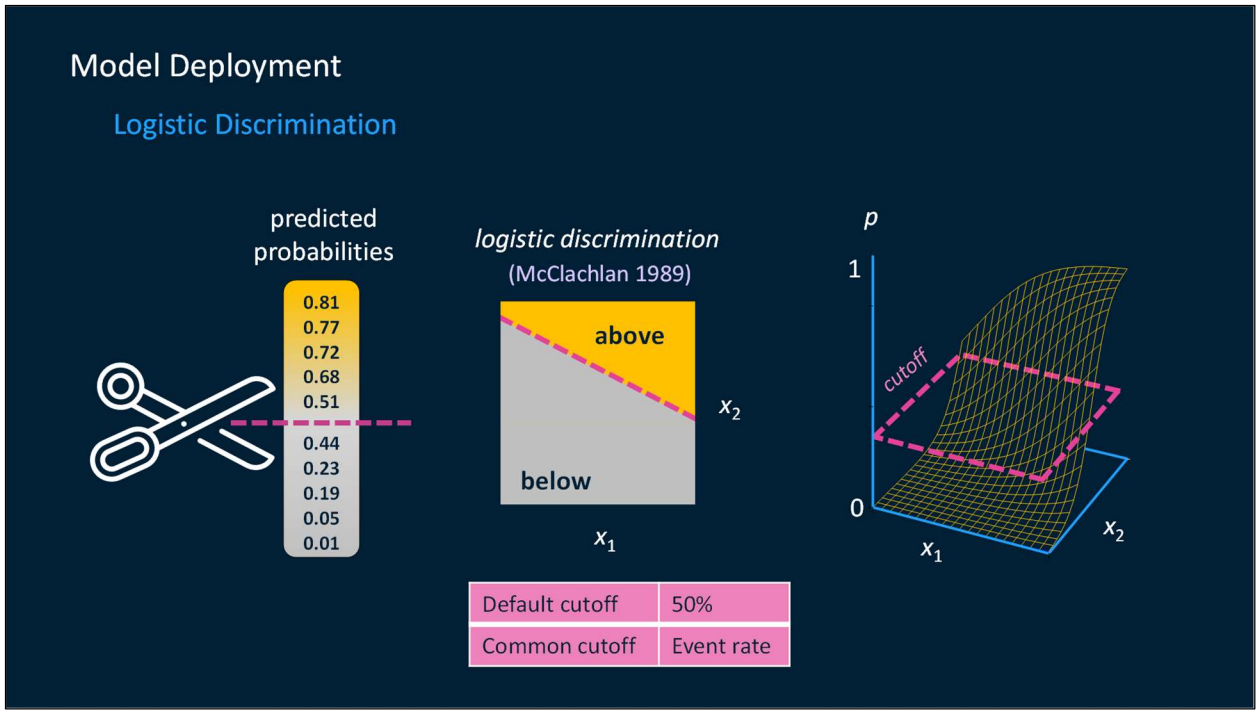
Predictions can be made by simply plugging in the new values of the inputs from the scoring data.

For scoring, the model is first translated into another format (typically, score code).

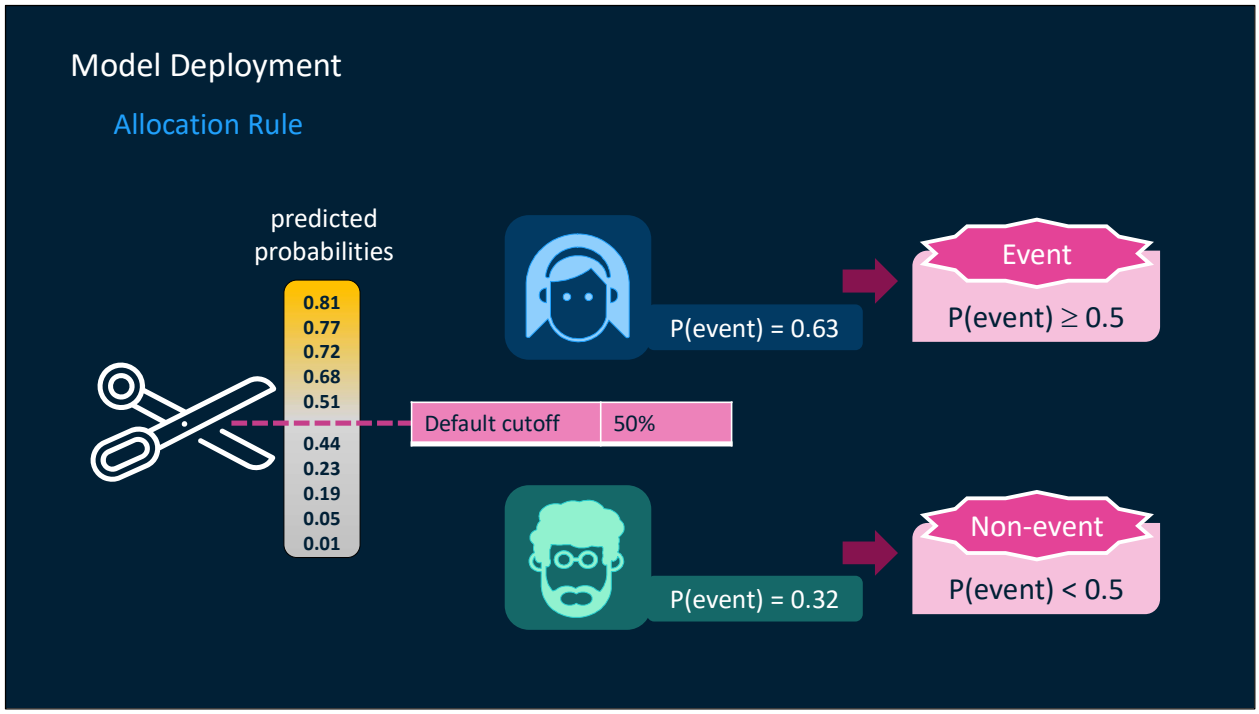
The logit equation produces a ranking or logit score.

You can obtain a prediction estimate using a straightforward transformation of the logit score, the logistic function. The *logistic function* is simply the inverse of the logit function. You can obtain the logistic function by solving the logit equation for *p*.

The most common predictions are the predicted probabilities.



In supervised classification, the ultimate use of logistic regression is to allocate cases to classes. This is more correctly termed logistic discrimination An allocation rule is merely an assignment of a cutoff probability, where cases above the cutoff are allocated to class 1 (event) and cases below the cutoff are allocated to class 0 (non-event). The standard logistic discrimination model separates the classes by a linear surface ((hyper)plane). The decision boundary is always linear. Determining the best cutoff is a fundamental concern in logistic discrimination. A logical cutoff is 50%, which is the default in SAS Viya, but typically a much better way to choose a cutoff is to use the percentage of events in the original data.



Thus, assuming a default cutoff of 50%, if your model predicts an event probability of 0.63 for someone, she would be predicted as an event, because the predicted probability is greater than 0.5. Whereas if someone is predicted with an event probability of 0.32 by your model, he would be predicted as a non-event, because the predicted probability is smaller than 0.5.

What Have You Learned?

- The goal of explanatory modeling is causal *explanation*. The goal of predictive modeling is *empirical prediction*.
- After a model and allocation rule are determined (training), the model must be applied to new cases (scoring).
- For scoring, SAS Viya translates the model into a format called *score code*, which is a SAS program.
- Scoring must incorporate all data manipulation tasks done before generating the model.

What Have You Learned?

- Predictive modeling typically involves choices from among a set of models. These might be different types of models, or they might be different complexities of models of the same type.
- Selecting model complexity involves a trade-off between bias and variance. Both bias and variance are forms of prediction error in machine learning.
- An overly simple model might not be flexible enough and generally leads to underfitting. An overly complex model might be too flexible and generally leads to overfitting.
- An optimal balance of bias and variance would never overfit or underfit the model.

What Have You Learned?

- An association exists between two categorical variables if the distribution of one variable changes when the value of the other variable changes.
- Chi-square test is commonly used for feature selection in machine learning.
- You use the odds ratio to measure the strength of the association between a binary predictor variable and a binary response variable.
- A logistic regression model applies a logit transformation, or simply the log odds transformation, to the probabilities.
- The c value is one of the most used measures of model performance.
- You need to determine the best cutoff in logistic discrimination.