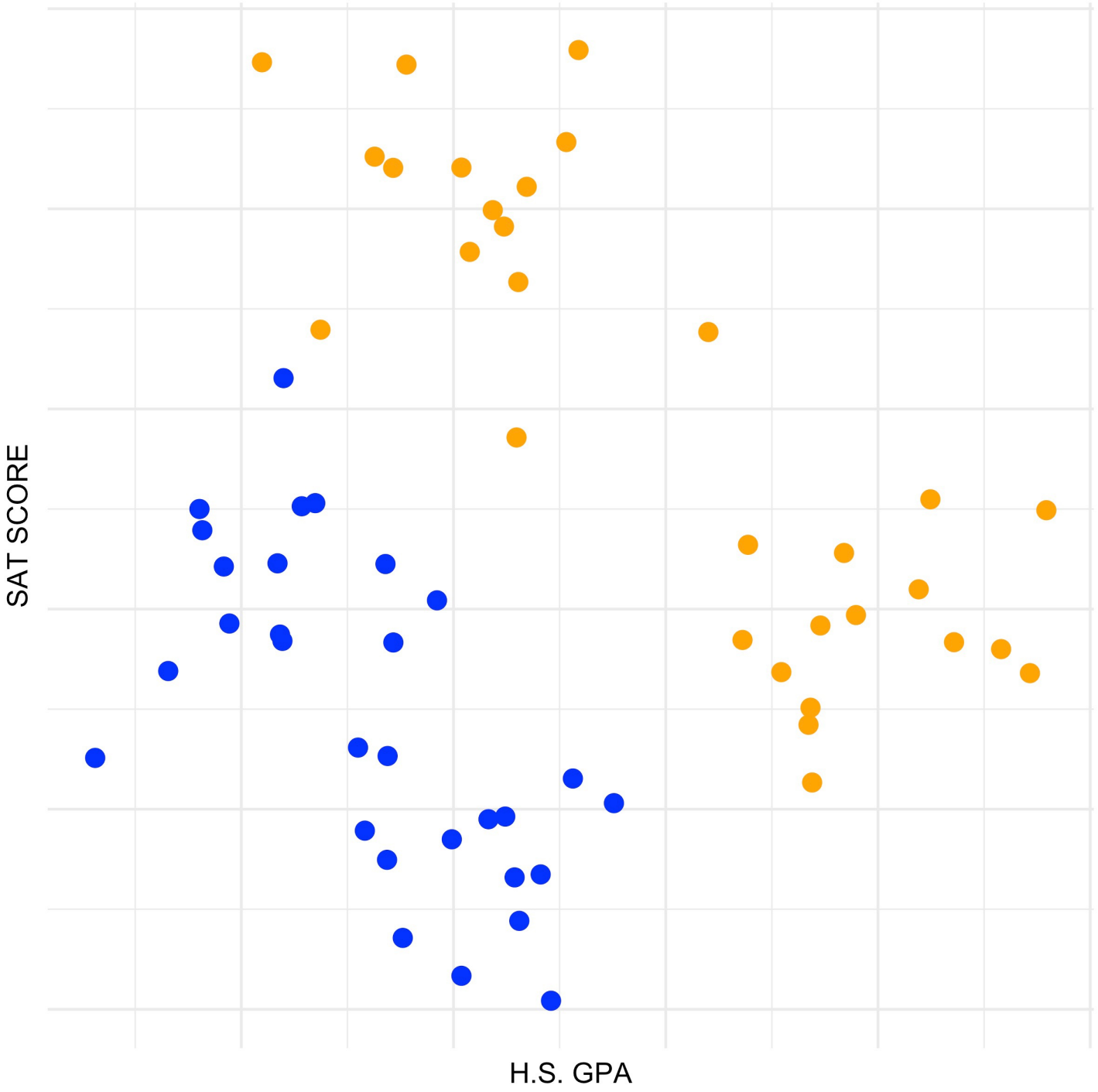
College Acceptance

Let's start with a relatively simple prediction task, where we try to predict whether a student will get accepted to UR based on their high school GPA and SAT score.

We have data for 60 students. Blue represents rejection and orange represents acceptence. Note: this data is completely fake!

How well do you think you can predict whether a student will be accepted based on these two variables?



College Acceptance

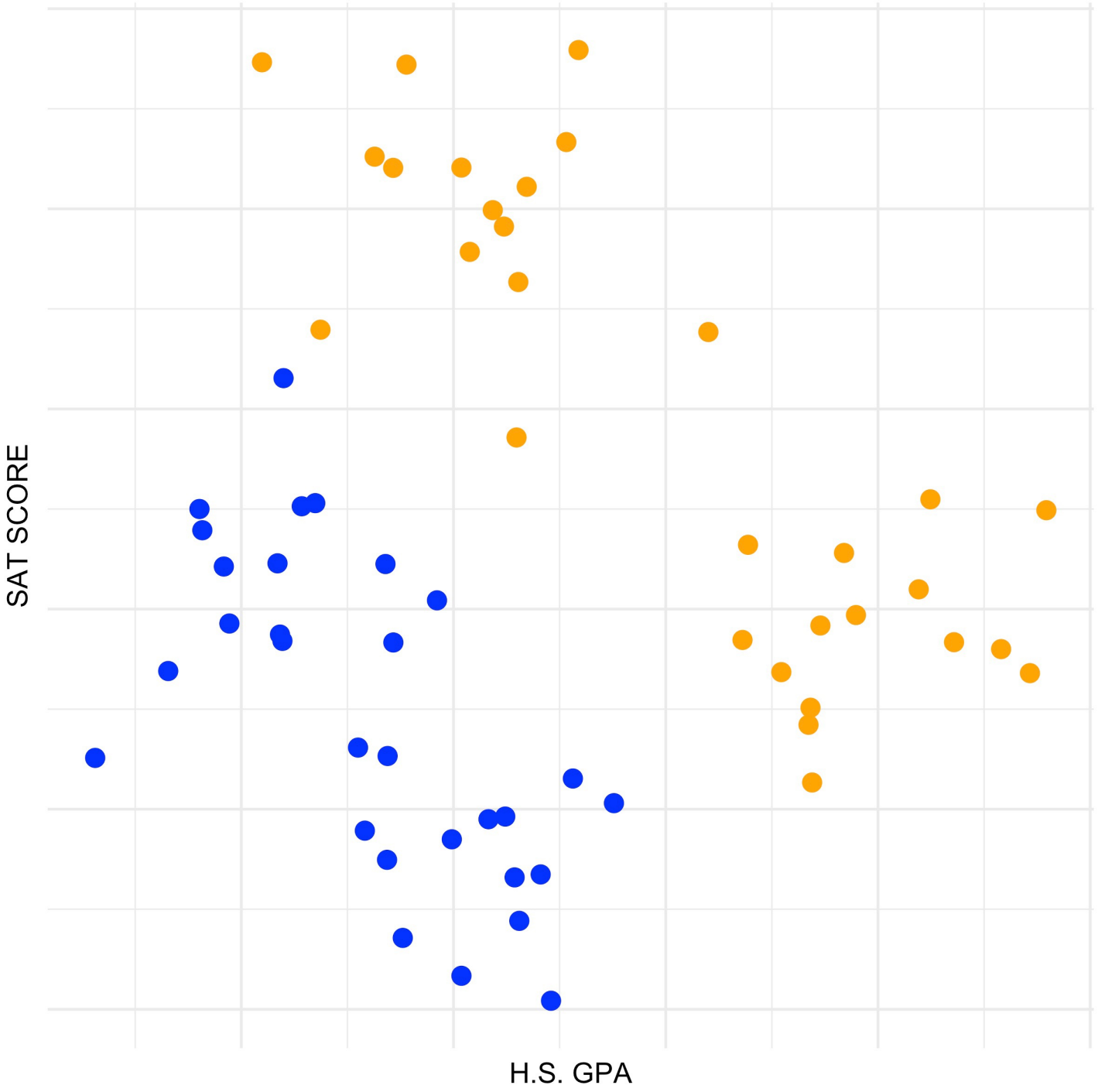
Before going any further, let's get some terminology down:

Observation: one data point that we can use to build a model; here each observation is a student

Features: the measurements the model used to make predictions; here we have two features: H.S. GPA and SAT scores

Label: these are is the thing we are trying to predict; each student has one label

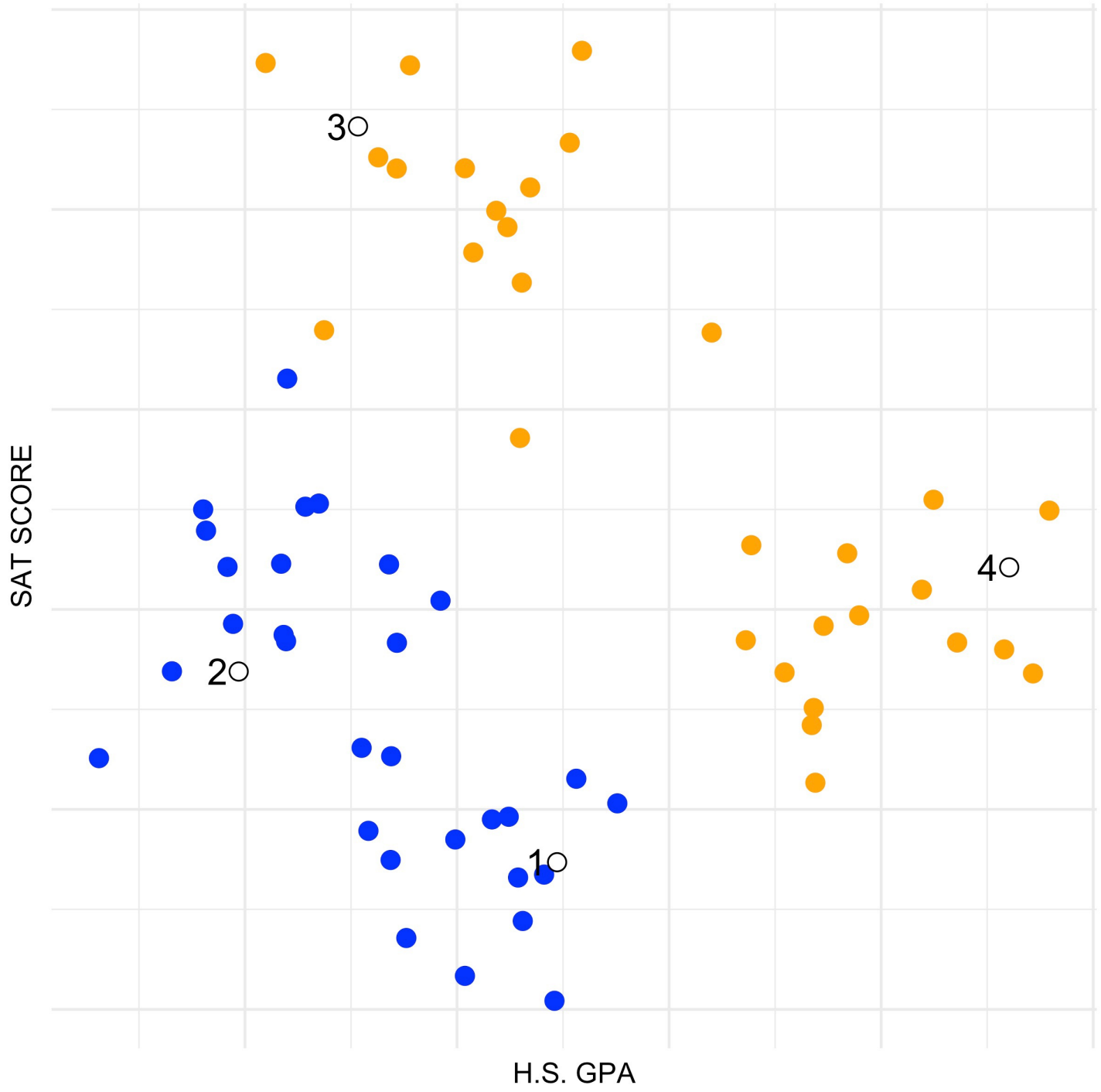
Classes: these are the possible values of the labels; here, the classes are accepted or rejected



Predict New Data

Now we will add four students to the dataset, but not reveal whether they were accepted to UR.

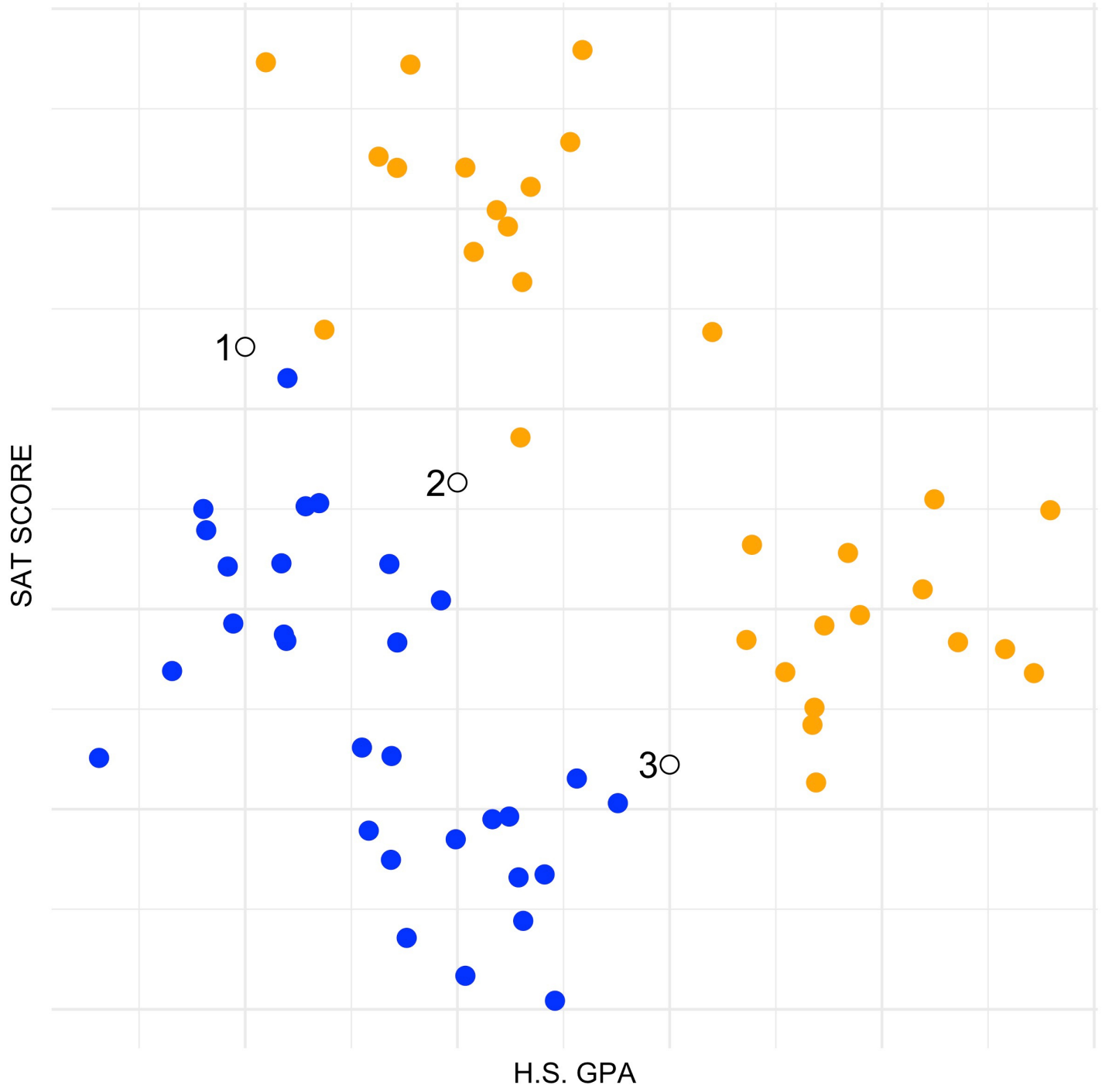
Without any fancy models, what would you predict for each of these students? Why? Are you relatively confident about these predictions?



Predict New Data

Now, consider these three students.

What you predict if you had to guess whether they would be accepted to UR? How confident do you feel about these predictions?

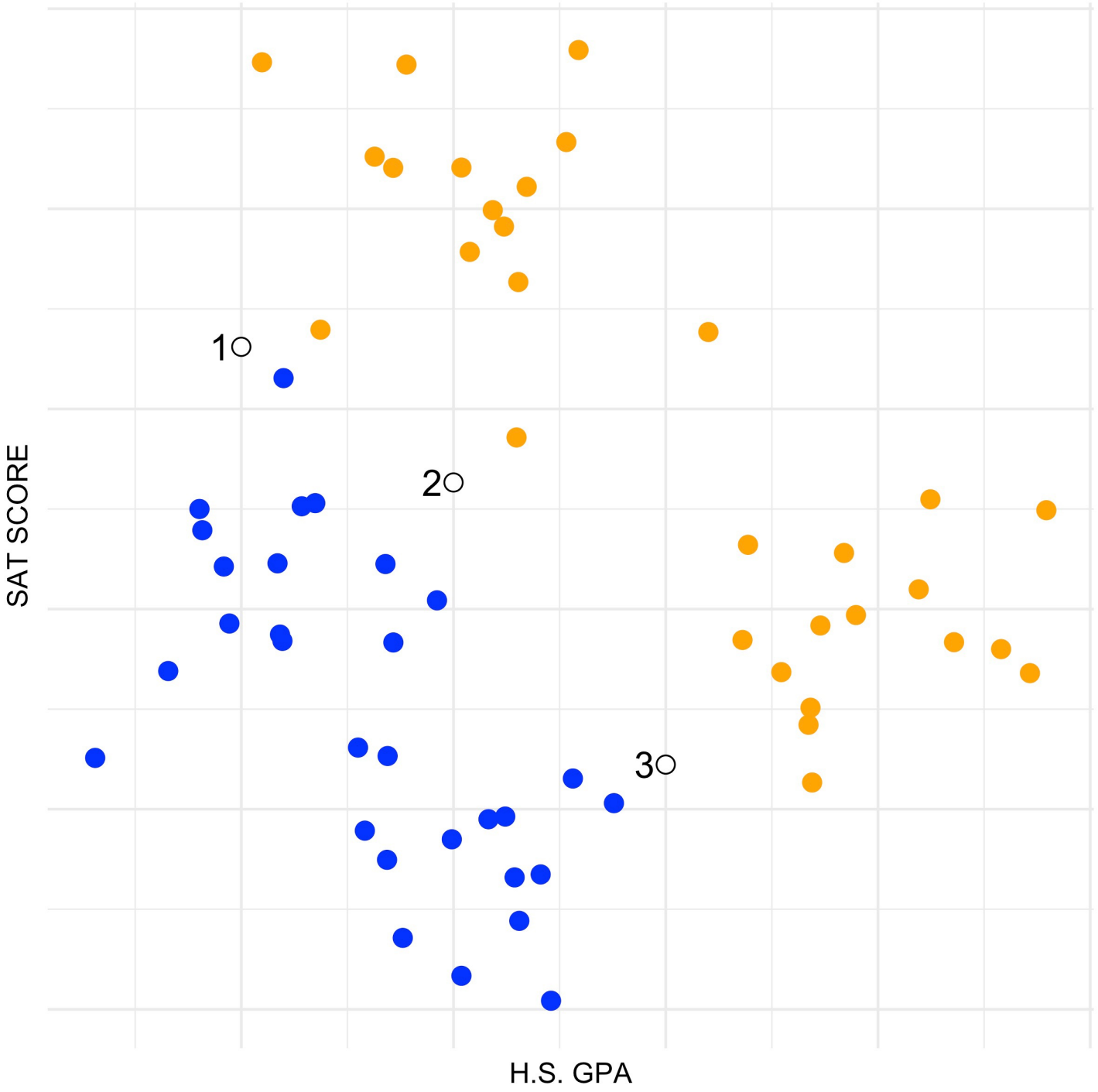


Predict New Data

Now, consider these three students.

What you predict if you had to guess whether they would be accepted to UR? How confident do you feel about these predictions?

These values are always going to be difficult to predict. But, to be able to make any reasonable guess, it would seem like we might need a more formal model.

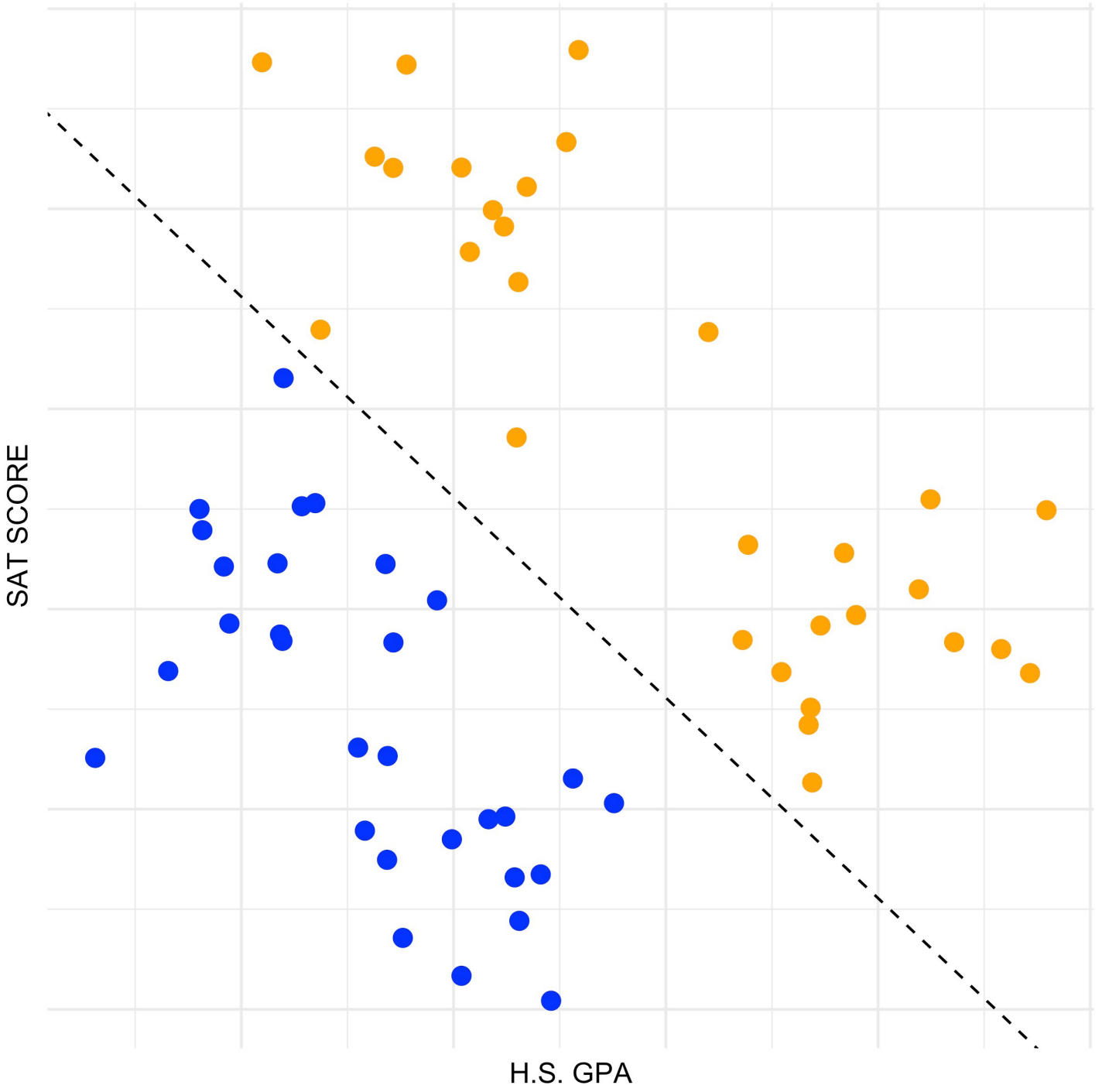


Classification Line

One of the most versitile methods for building a predictive model is to find a way to split the data by using a line. We expect points on one side of the line to be of one label and points on the other side of the line to be of a different label.

I have drawn a line on our plot. Does it seem to reasonably seperate the two labels?

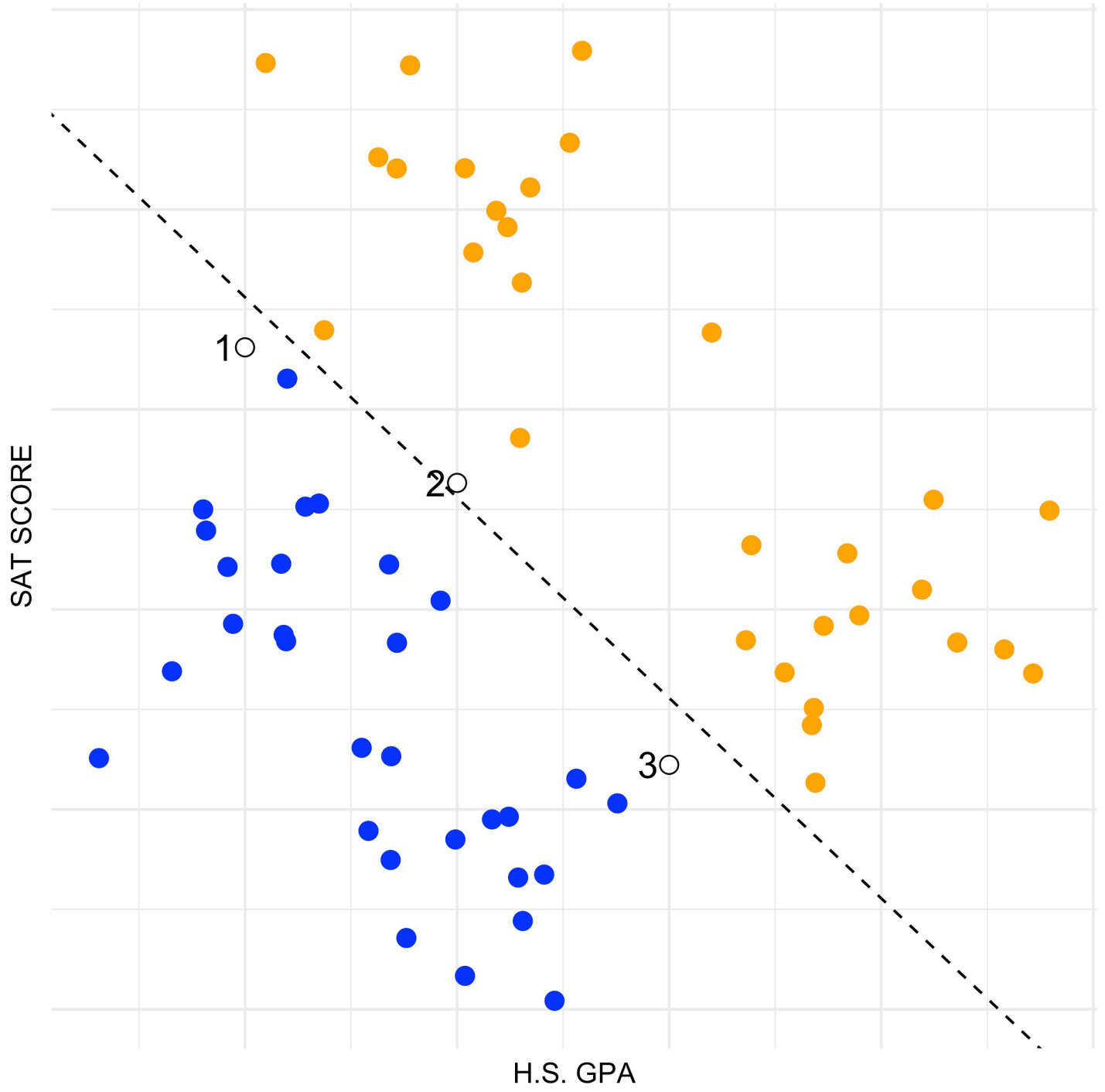
We will come back in a moment to discuss the specific process for constructing a good seperating line.



Classification Line

Now with our line we can make predictions for each of the difficult points.

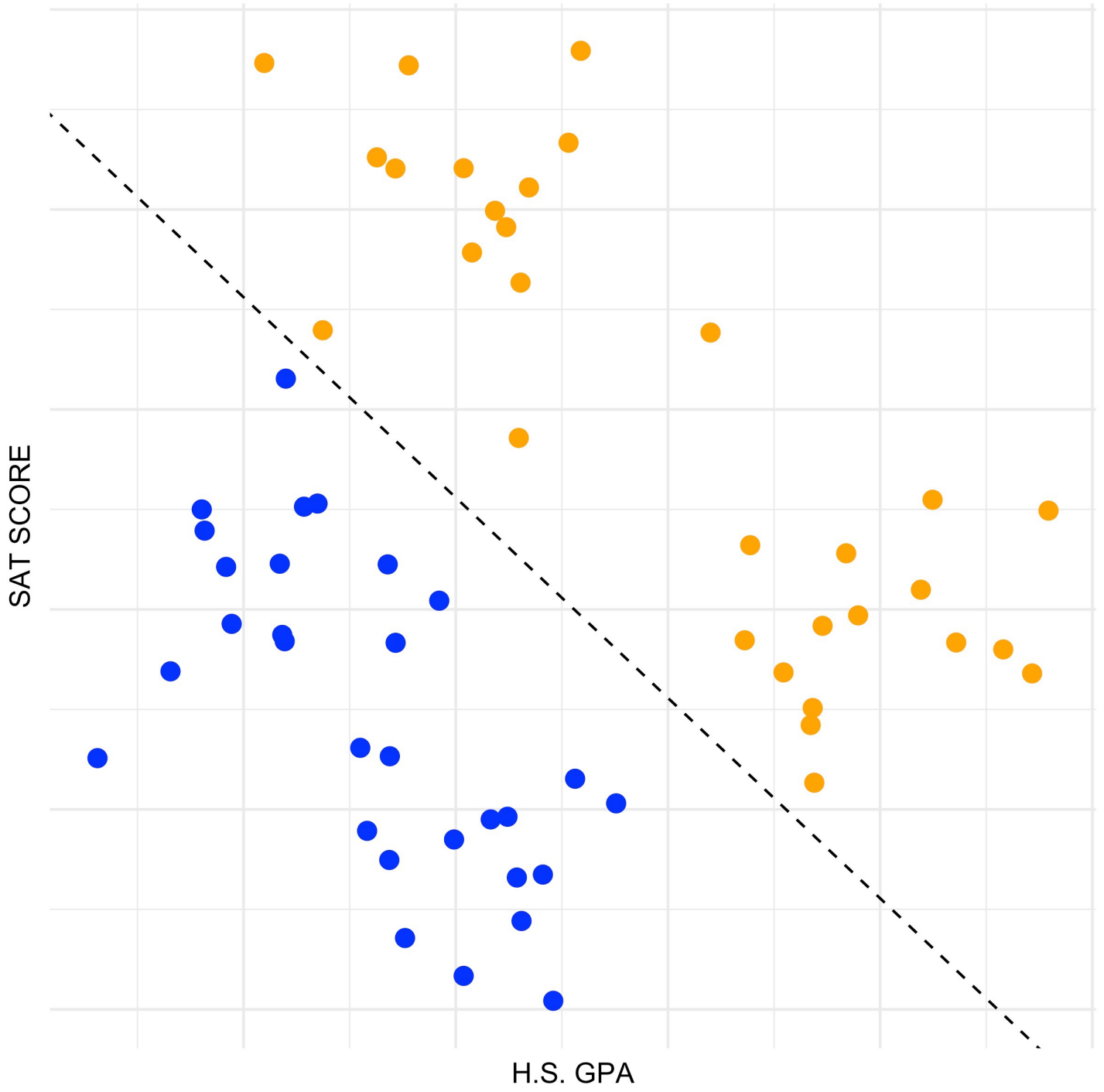
What label does our model predict for each of the three students for which we do not have labels? How confident are you in these predictions?



Model Evaluation

One important task in prediction modelling is to evaluate how well our model is able to make predictions on new values. We can do this by using the **error rate**, which is simply the percentage of predicted labels that were incorrect.

In the model we built on the right, our error rate is 0%. It's perfect! But, we are cheating a bit because we are evaluating the model with the same data that was used to build the model.



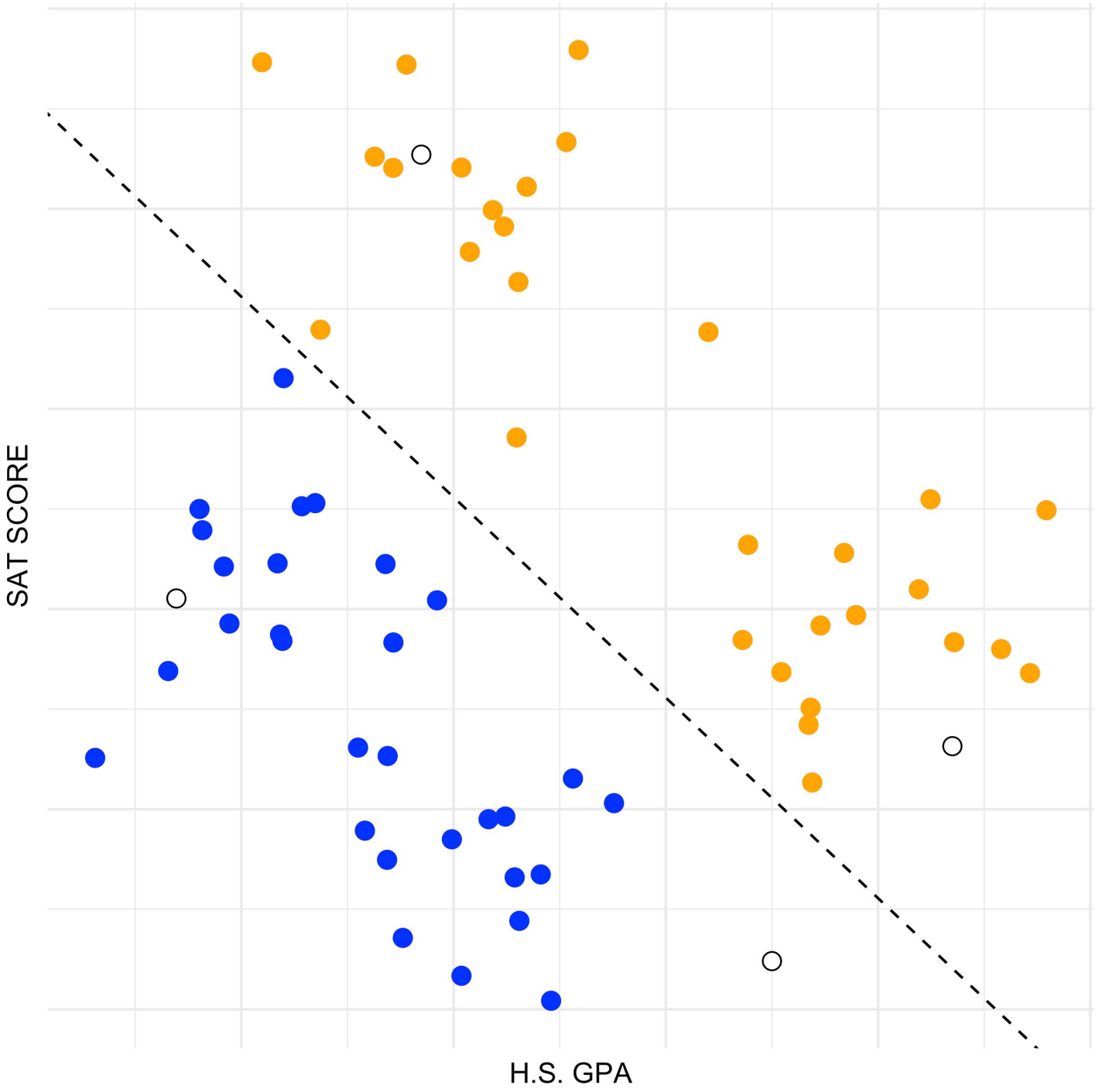
Model Evaluation

The most common approach to this issue is to randomly split our data into two parts:

Training data: data used to build a predictive model

Validation data: data used to evaluate how well the model works on new data

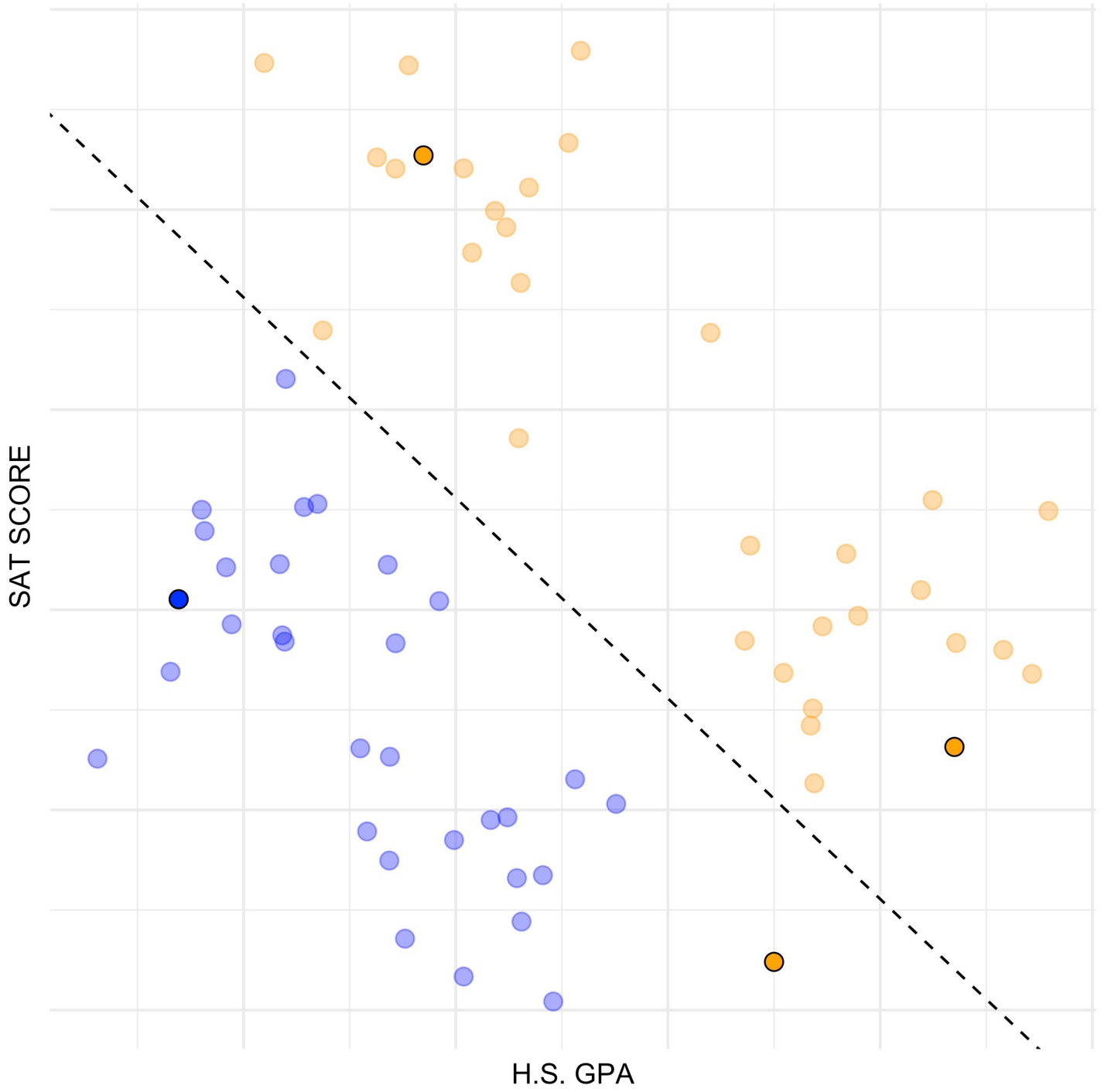
In the plot to the right, we have four validation points defined by the white circles. These are not used when defining the classification line.



Model Evaluation

Now, we can evaluate the model. Note that it made three correct predictions and one incorrect prediction.

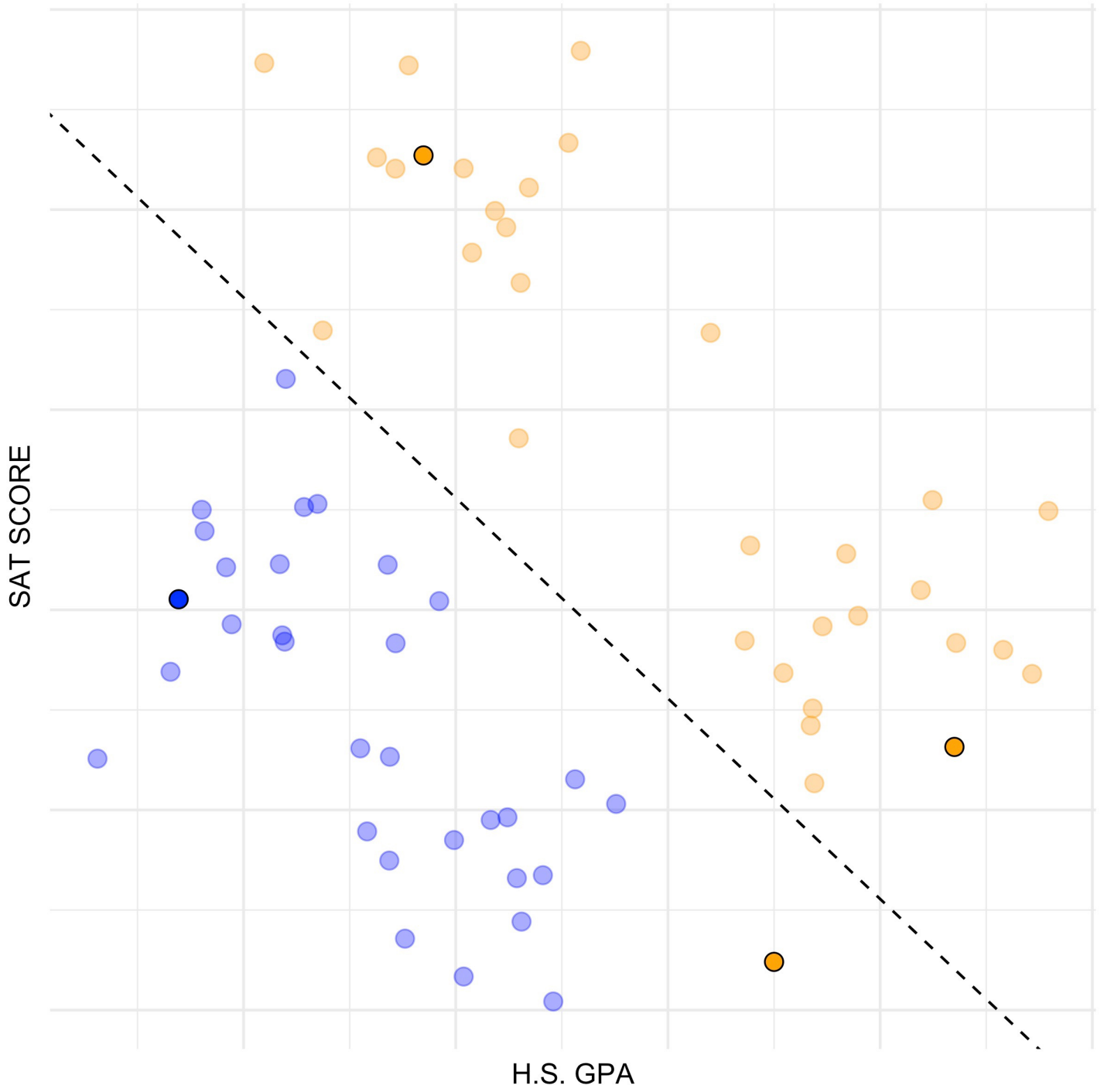
The model has a **training error rate** of 0% and a **validation error** rate of 25%.



Confusion Matrix

Another way to evaluate a model is to build a **confusion matrix**. This gives a more grainular view of the errors that occur on the validation data. It is easiest to understand with an example:

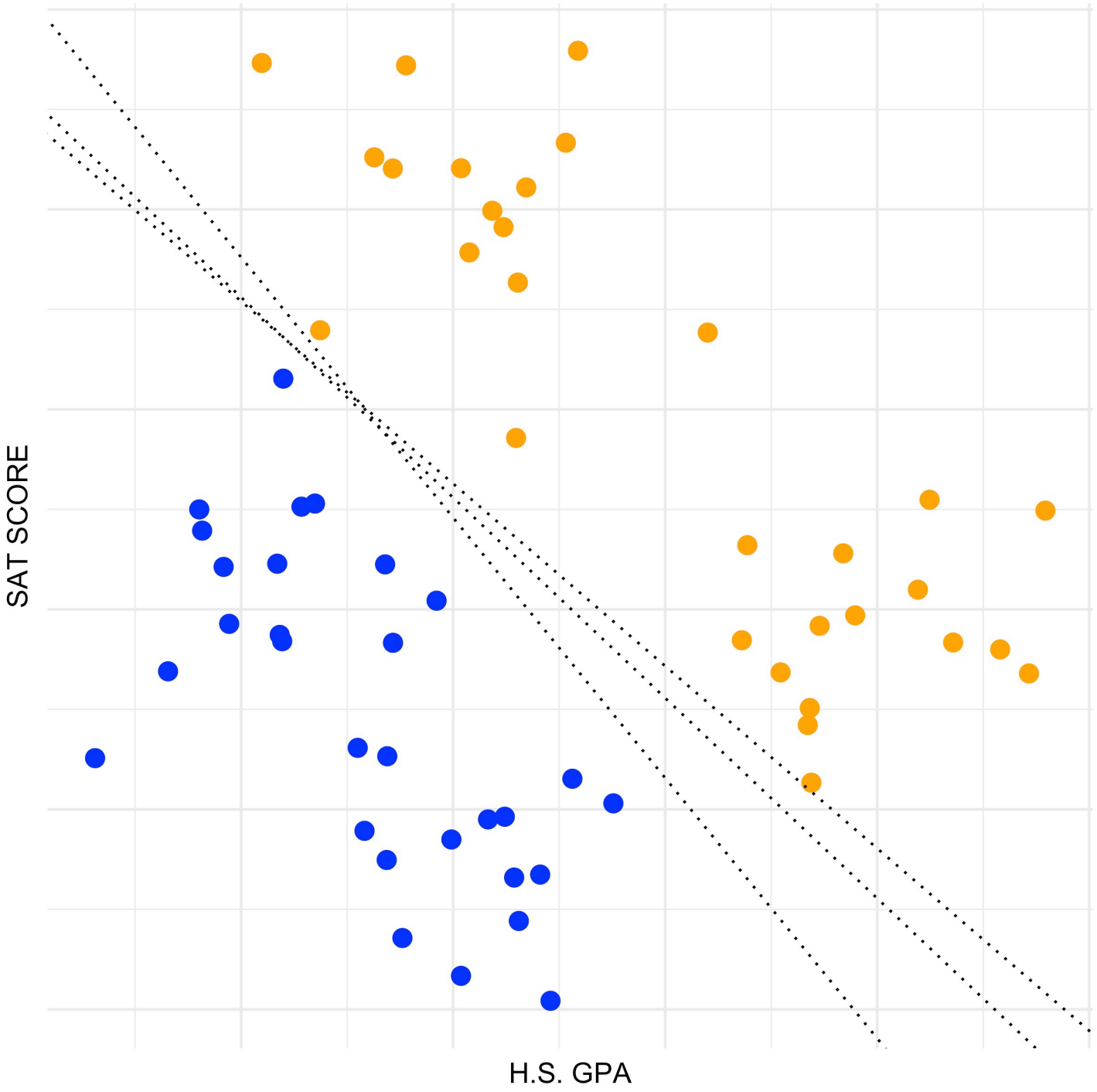
		PREDICTED	
		1	0
ACTUAL		1	2



Choosing a Line

Let's return to the task of building a model from training data.

Previously, I drew a line and we agreed that it seemed reasonable because it had an error rate of zero on the training data. However, there are many lines that have an error rate of zero on this training data. Example are given on the right.

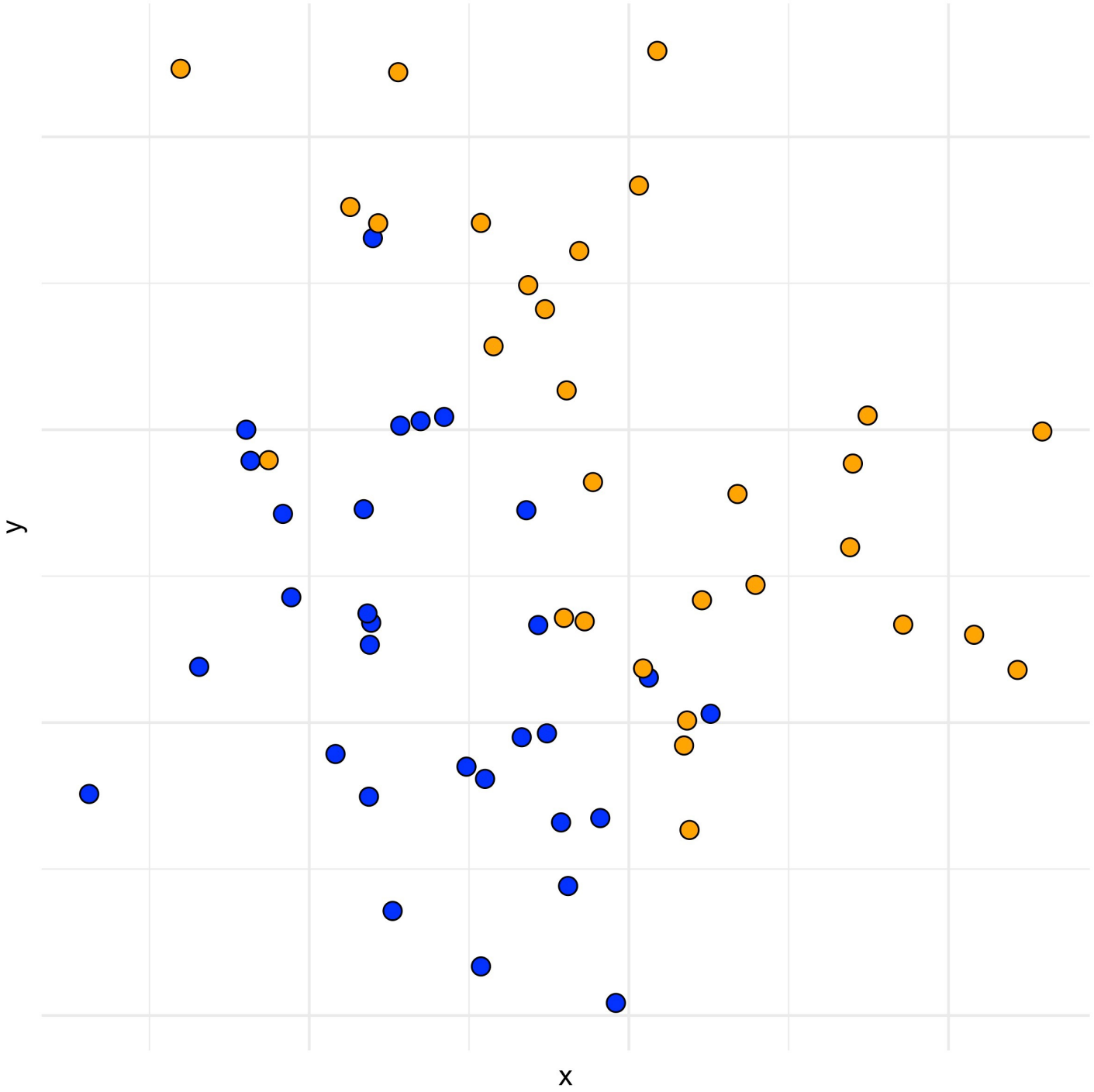


Choosing a Line

Consider a similar task, but with some different data. We will move an abstract features called x and y. You can still think of the college acceptance example if you prefer something concrete.

Notice that in this case there is no line that perfectly seperates the two label types.

We could choose a line based on one that makes the fewest errors, but there will again be many different choices that we need to choose between.



Probability

A probability is a number **between 0 and 1** that gives the proportion of times we would expect an event to happen if we could replicate observing something many times.

It turns out that the best approach to building a classification line is to consider the slightly more complex task of predicting the probability that a point is equal to one of the two classes.

Probability Model

Let's define a model with three **parameters**, numbers that define the model, according to the equation to the right.

The values **a**, **b**, and **c** can be set to any real numbers we want. The idea is that we will use the training data to determine good values for them and then can use the model to predict values on the validation data (or even, entirely new data that we do not yet have).

$$\text{Probability}(\bullet) = a + b \times X + c \times Y$$

Classification Line

Points with a probability greater than 0.5 are those where we think that the point is likely to be orange.

With some simple algebra, we can re-arrange to see that (assuming **c** is positive),* this defines a line with a slope and intercept.

So, we see that this model defines a linear classification line!

** When **c** is negative, the inequality reverses, which still makes a line, but with the orange points below rather than above. A value of zero gives a horizontal line; still okay, but some of the math gets a bit messy.*

$$\text{Probability}(\bullet) = a + b \times X + c \times Y$$

$$0.5 < a + b \times X + c \times Y$$

$$0.5 - a - b \times X < c \times Y$$

$$(0.5 / c - a / c) - (b / c) \times X < Y$$

$$Y > (-b / c) \times X + (0.5 / c - a / c)$$

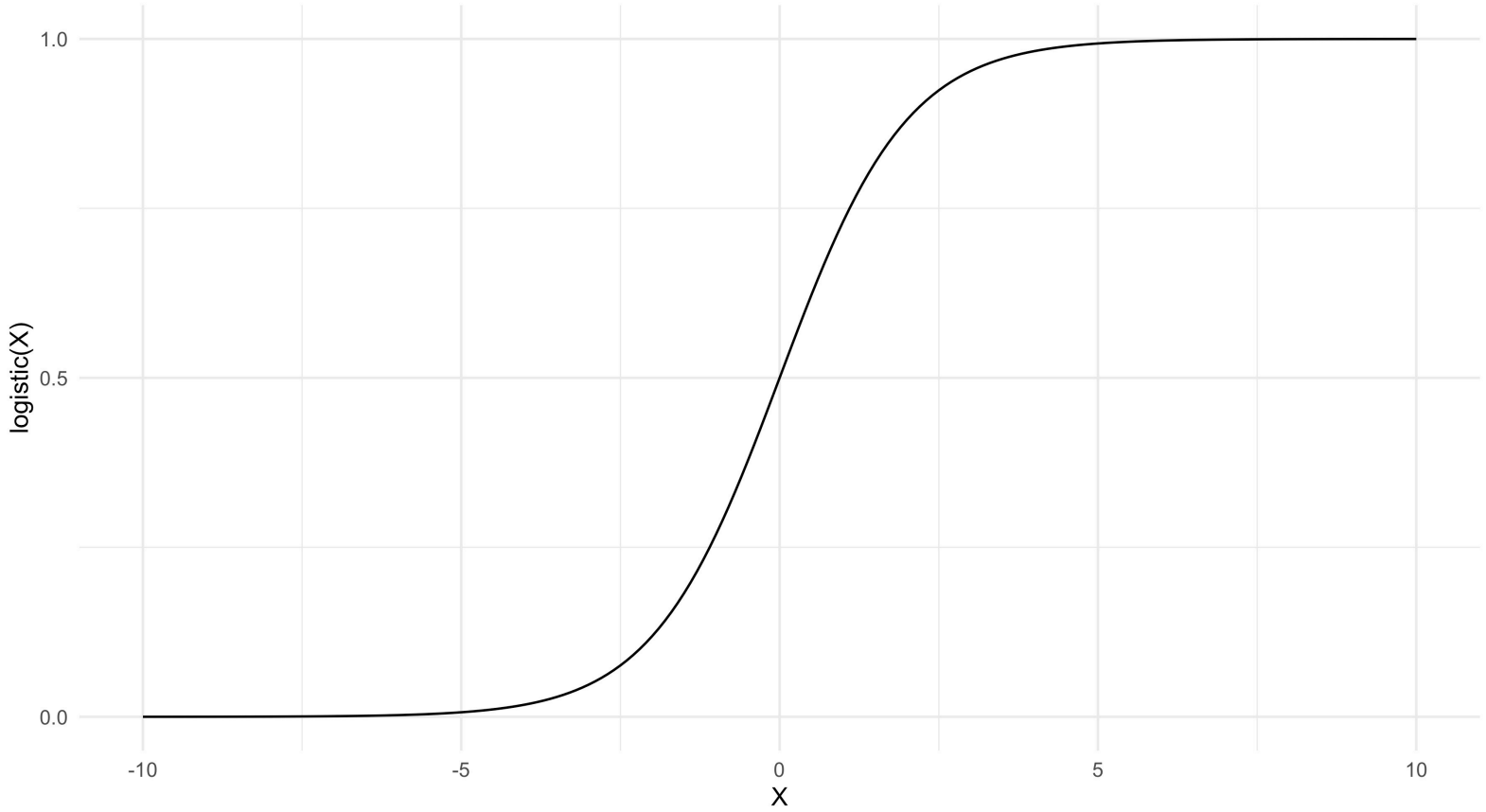
$$Y > [\text{slope}] \times X + [\text{intercept}]$$

One Tweak

There is one issue with our model: it is possible that we could produce probabilities that are less than zero or greater than 1. This can be fixed fairly easily by adding a **link function** that maps any real number to a number between zero and one.

The most common choice is the logistic function: exp(x) / (exp(x) + 1). A visualization of the function is shown on the right.

Probability(●) = F(**a** + **b** × X + **c** × Y)



Classification Line (again)

This does not actual affect our primarily conclusion before that the model creates a classification line. The only different is that now we want to find points where the value of (a + b × X + c × Y) is 0 instead of 0.5.

$$\text{Probability}(\bullet) = F(a + b \times X + c \times Y)$$

$$0 < a + b \times X + c \times Y$$

$$0 - a - b \times X < c \times Y$$

$$(a / c) - (b / c) \times X < Y$$

$$Y > (-b / c) \times X - (a / c)$$

$$Y > [\text{slope}] \times X + [\text{intercept}]$$

Learning Parameters

Okay great, but we still have not explain how to learn the values a, b, and c from the training data.

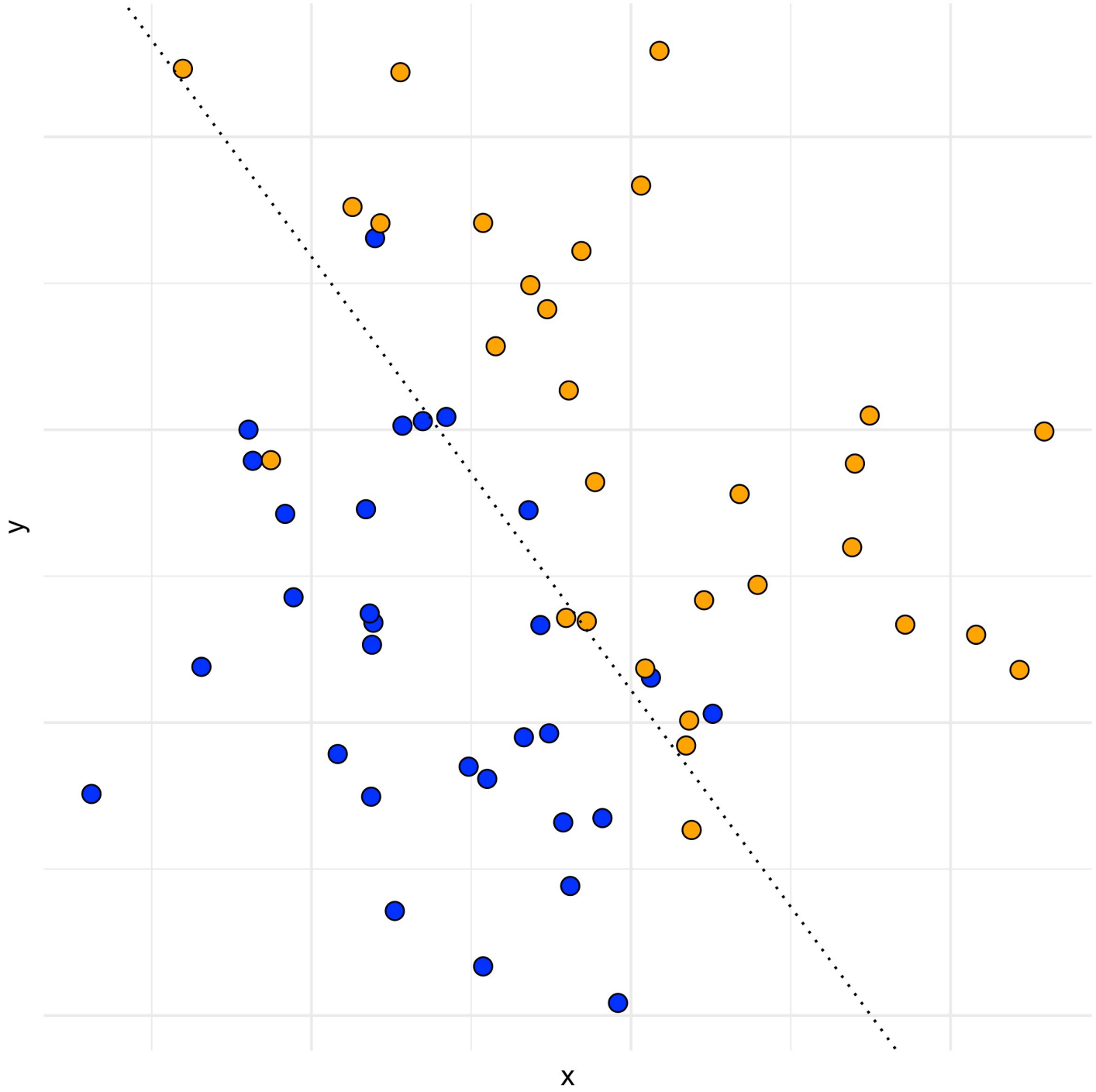
The trick is to notice that our model now gives specific probabilities that each point is equal to a particular class. What we can do then is find the values of a, b, and c that **maximize the probability** of observering the training data.

Because we have *continuous probabilities*, this will almost always have a unique solution. We won't go through all of the math here; there is no analytic solution, but it can be solved using fast numerical methods.

$$\text{Probability}(\bullet) = F(a + b \times X + c \times Y)$$

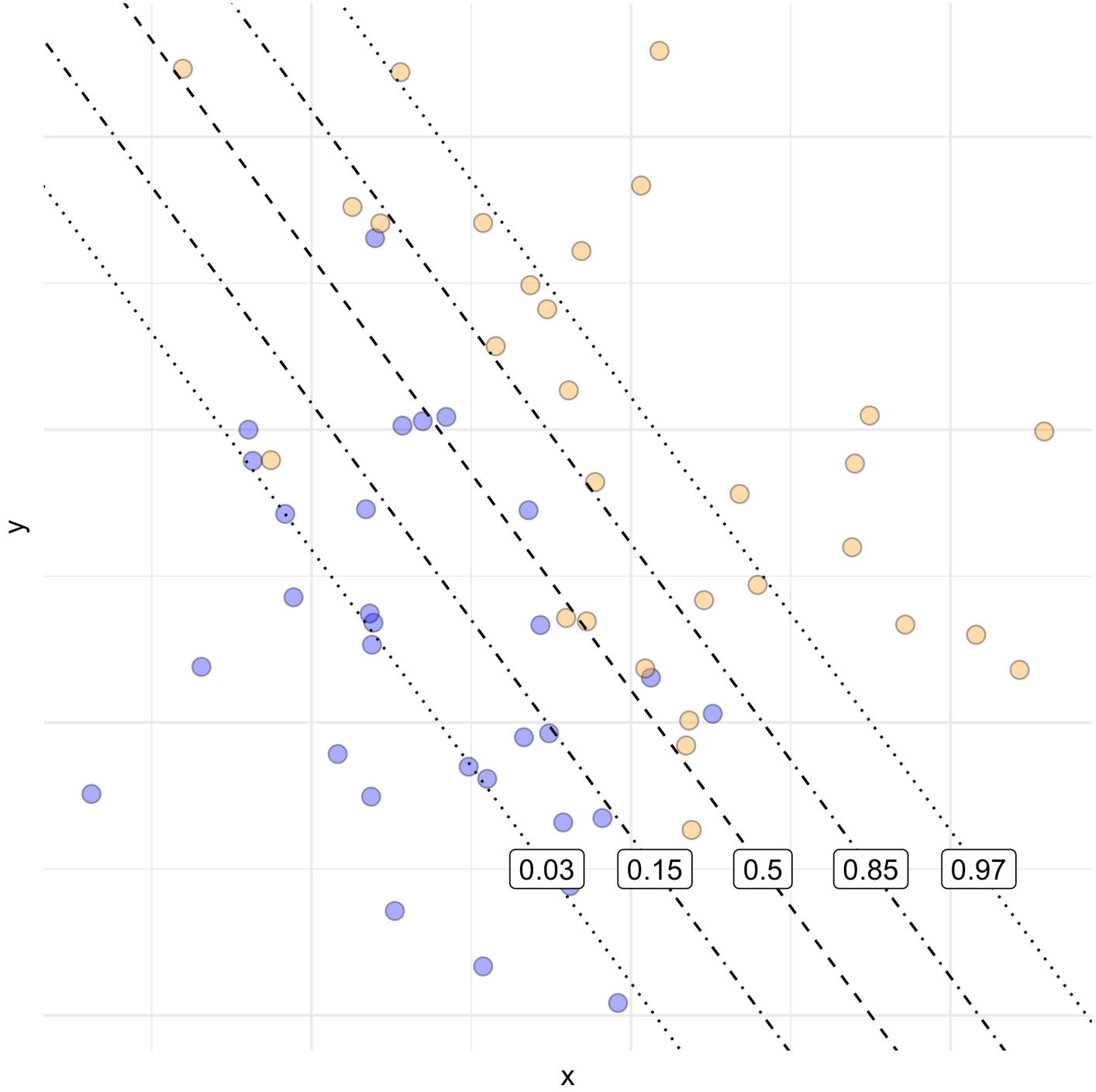
Prediction

Let’s go back to our data and look at the line defined by the model. Note that it seperates the data well but not perfectly.



Prediction

Because we have probabilities, we can actually plot different contours of probabilities. Each of these will be a parallel line. Notice how quickly the probabilties drop off as we move away from the classification line.



Interpret Parameters

$$\text{Probability}(\bullet) = F(a + b \times X + c \times Y)$$

Because of the transformation by the link function and the relative scales of X and Y, it is can hard to directly interpret the exact meaning of the values **a**, **b**, and **c**. However, we can say something things:

– If **b** is positive, X appears to be positively related to the orange class.

– If **b** is negative, X appears to be negatively related to the orange class.

– If **b** is zero (or very small), X appears to have little to no effect on an observation's label.

The same things can be said with repect to **c** and the variable Y.

Wrap-Up

We have covered a lot today. The method we defined is called **logistic regression.** It is perhaps the most fundamental method in all of machine learning.

Make sure to review these notes before the next class!