Module 3-1: Intro to Classification

So you've mastered regression and know how to train a model. But what if the predictions you want to make are not necessarily numerical? What if you wanted your model to predict if an animal is a dog or a cat depending on paw size, fur pattern, and weight? What if you wanted your model to predict if a new email was spam or not depending on the contents of the email?

A regression model would struggle to make these kinds of predictions. All it gives you are numbers. However, a classification model can do these quite intuitively. Today we will be discussing classifications, its uses, as well as a few different classification algorithms that `sklearn` provides.

Classification is a machine learning technique where a model will learn how to assign a provided class label to provided data. The important part is to keep in mind is that the classes are pre-defined by the user.

Linear Classifiers

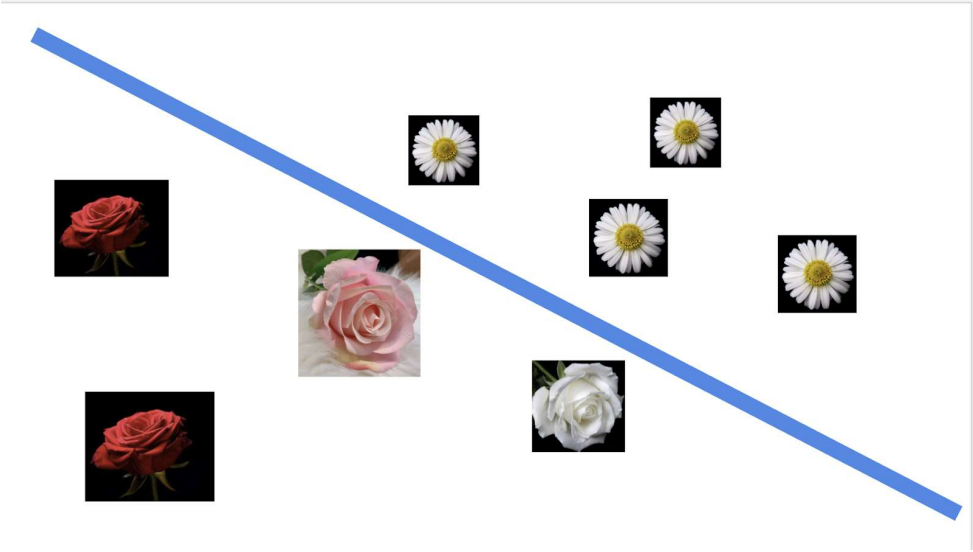
A **linear classifier** is a type of classifier that "splits" a data set into two groups. Imagine (for simplicity) that we laid out our data onto a table. A classifier would draw a line on the table to split the data in two. If you give it a new piece of data, it either places it on the left or right side.

Let's take a look at an example of a linear classifier. Using this data, we want to train a linear classifier that labels a flower as a rose or a daisy depending on its color.

Send Issue

Petal Color	Flower Type
White	Daisy
White	Daisy
Red	Rose
Pink	Rose
Red	Rose
White	Rose
White	Daisy
White	Daisy

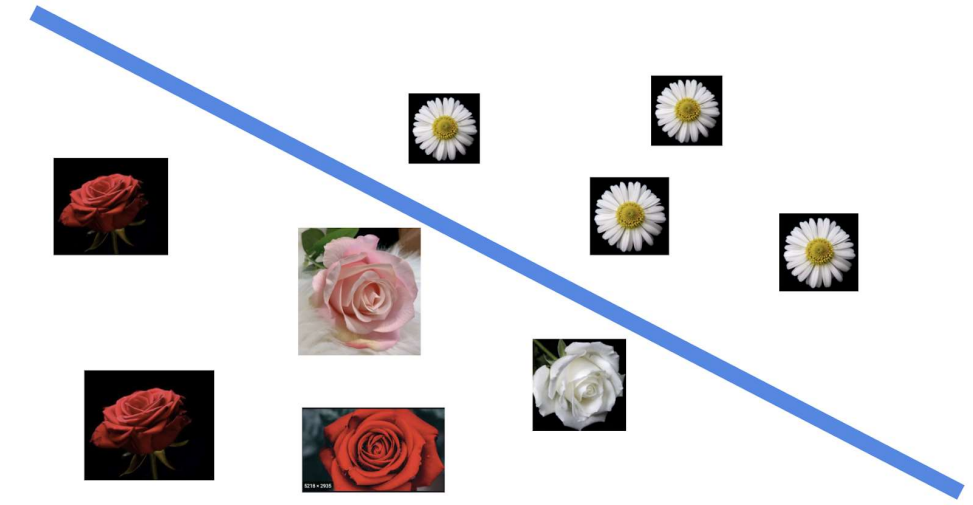
Conceptually, we want our trained classifier to do something like this, where it can split flowers apart from one another. The blue line in the middle represents the linear classifier.



So far, all it did was read the table and place a flower on the left if it was a rose and on the right if it was a Daisy.

Now, let's try to predict the type of a flower that the model has never seen before. Let's try to predict a red rose flower color. Our freshly trained linear classifier looks at the two classes and determines that a red flower belongs on the left side with the other red flowers, and therefore it must be a rose.

This is an example of where our linear classifier might place the new red flower. Since it is to the left of the blue line, it means that the classifier thinks that the flower is a rose.



However, there are some issues with our current model! This model is not very sophisticated because we did not use a lot of data to train it. We only used eight total data points as well as only one

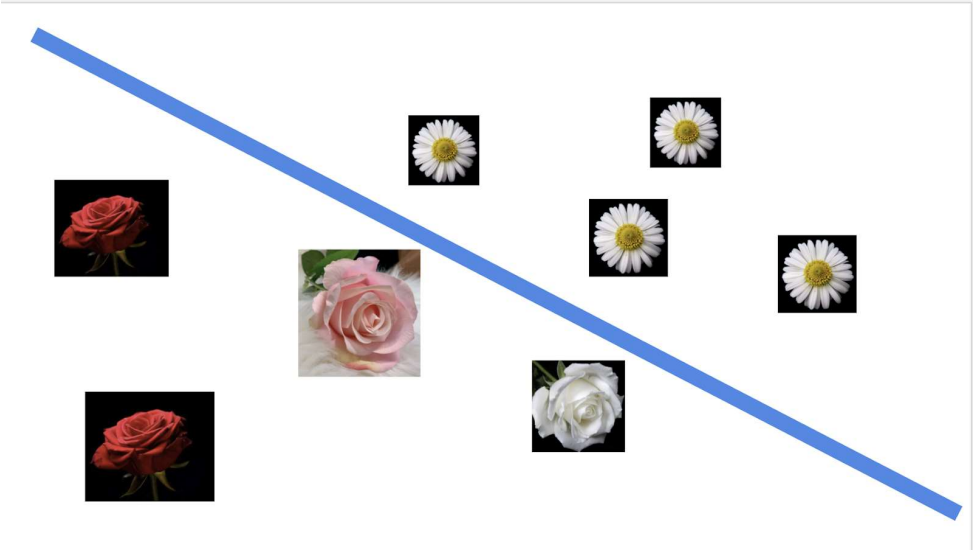
feature (the petal color) to classify which type of flower it was. This means that it can get confused and predict very easily.

Say we want to predict a white rose. We run into an issue, since our model only uses flower color to determine what kind of flower that will be predicted. Since all of the daisies are white, it might wrongly predict a white rose as a daisy.

However, this can be solved by adding more features to our model to make it more adaptable. We can train our model again with the following added feature. The thorns column has a 0 if the flower does not contain any thorns. The thorns column has a 1 if the flower contains any thorns.

Petal Color	Flower Type	Thorns
White	Daisy	0
White	Daisy	0
Red	Rose	1
Pink	Rose	1
Red	Rose	1
White	Rose	1
White	Daisy	0
White	Daisy	0

Now that our model has more features it has more values to base its predictions off of, our new trained model will still look like this.



All the flowers on the left are roses and have thorns. All the flowers on the right are daisies and don't have thorns.

If we were to have our model predict a white flower with thorns, it could properly place it as a rose because it knows that none of the daisies have thorns. Conversely, if our model was going to predict a white flower without thorns, it would properly predict it was a daisy because none of the daisy have thorns.

It is always important to train your data with an appropriate amount of features so that you can get optimal results. Additionally, you want to make sure that you have much more than just 8 data points. That way you can cover more edge cases than the example, such as roses that have different colors.

This can apply to all sorts of data! Try to classify the dogs below and discuss how they may be classified.