**Introduction**

When an email lands in your inbox, how does your email service know whether it's a real email or spam? This evaluation is made billions of times per day, and one way it can be done is with Logistic Regression. ***Logistic Regression*** is a supervised machine learning algorithm that uses regression to predict the continuous probability, ranging from 0 to 1, of a data sample belonging to a specific category, or class. Then, based on that probability, the sample is classified as belonging to the more probable class, ultimately making Logistic Regression a classification algorithm.

In our spam filtering example, a Logistic Regression model would predict the probability of an incoming email being spam. If that predicted probability is greater than or equal to 0.5, the email is classified as spam. We would call spam the *positive class*, with the label 1, since the positive class is the class our model is looking to detect. If the predicted probability is less than 0.5, the email is classified as ham (a real email). We would call ham the *negative class*, with the label 0. This act of deciding which of two classes a data sample belongs to is called *binary classification*.

Some other examples of what we can classify with Logistic Regression include:

* Disease survival —Will a patient, 5 years after treatment for a disease, still be alive?
* Customer conversion —Will a customer arriving on a sign-up page enroll in a service?

In this lesson you will learn how to perform Logistic Regression and use it to make classifications on your own data!

If you are unfamiliar with Linear Regression, we recommend you go check out our [Linear Regression course](https://www.codecademy.com/paths/data-science/tracks/dspath-supervised/modules/dspath-linear-regression/lessons/linear-regression/exercises/introduction) before proceeding to Logistic Regression. If you are familiar, let's dive in!

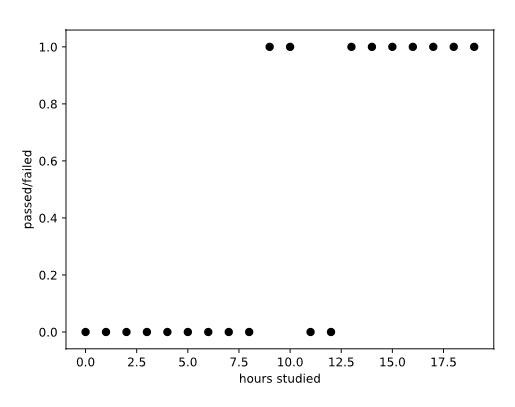
Instructions

**1.**

Codecademy University's Data Science department is interested in creating a model to predict whether or not a student will pass the final exam of its Introductory Machine Learning course. The department thinks a Logistic Regression model that makes predictions based on the number of hours a student studies will work well. To aid the investigation, the department asked a supplemental question on the exam: how many hours did you study?

Run the code in **script.py** to plot the data samples. 0 indicates that a student failed the exam, and 1 indicates a student passed the exam.   
  
How many hours does a student need to study to pass the exam?

import codecademylib3\_seaborn  
import numpy as np  
import matplotlib.pyplot as plt  
from exam import hours\_studied, passed\_exam, math\_courses\_taken  
  
# Scatter plot of exam passage vs number of hours studied  
plt.scatter(hours\_studied.ravel(), passed\_exam, color='black', zorder=20)  
plt.ylabel('passed/failed')  
plt.xlabel('hours studied')  
  
plt.show()



**Linear Regression Approach**

With the data from Codecademy University, we want to predict whether each student will pass their final exam. And the first step to making that prediction is to predict the probability of each student passing. Why not use a **Linear Regression** model for the prediction, you might ask? Let's give it a try.

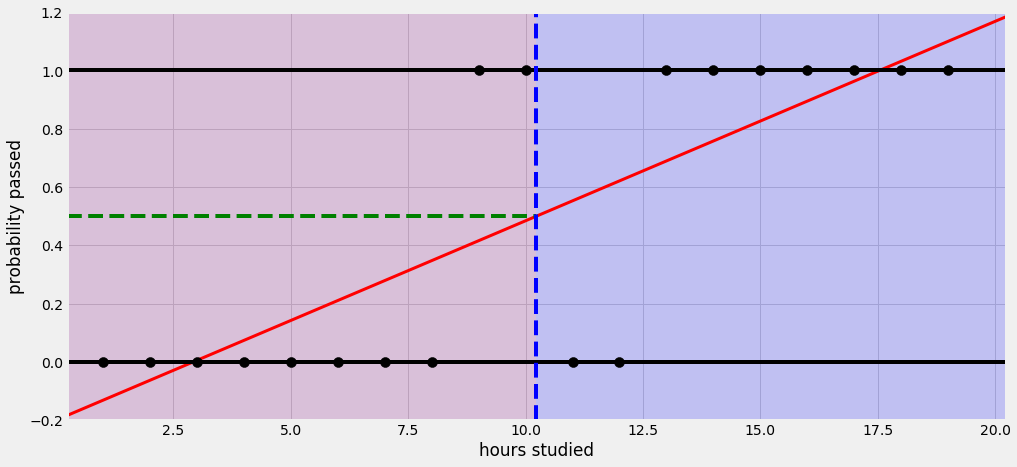
Recall that in Linear Regression, we fit a regression line of the following form to the data:

y=b0+b1x1+b2x2+⋯+bnxn

where

* y is the value we are trying to predict
* b\_0 is the intercept of the regression line
* b\_1, b\_2, … b\_n are the coefficients of the features x\_1, x\_2, … x\_n of the regression line

For our data points y is either 1 (passing), or 0 (failing), and we have one feature, num\_hours\_studied. Below we fit a Linear Regression model to our data and plotted the results, with the line of best fit in red.



A problem quickly arises. For low values of num\_hours\_studied the regression line predicts negative probabilities of passing, and for high values of num\_hours\_studied the regression line predicts probabilities of passing greater than 1. These probabilities are meaningless! We get these meaningless probabilities since the output of a Linear Regression model ranges from -∞ to +∞.

Instructions

**1.**

Provided to you is the code to train a linear regression model on the Codecademy University data and plot the regression line. Run the code and observe the plot. Expand the plot to fullscreen for a larger view.

Using the regression line, estimate the probability of passing for a student who studies 1 hour and for a student who studies 19 hours. Save the results to slacker and studious, respectively.

What is wrong with using a Linear Regression model to predict these probabilities?