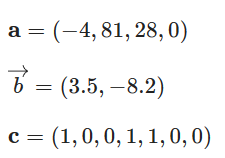
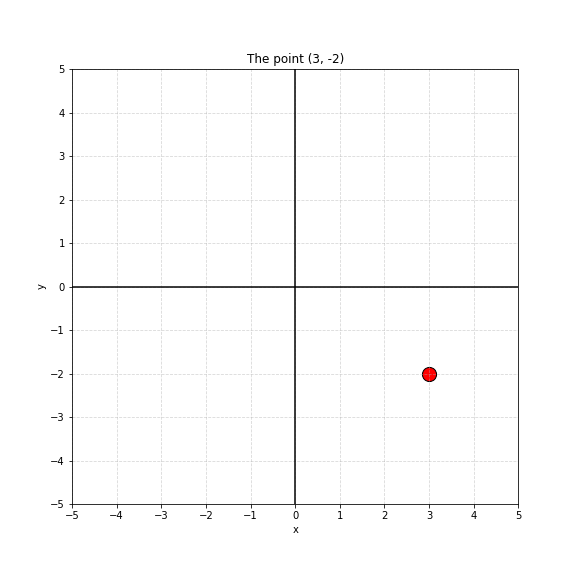
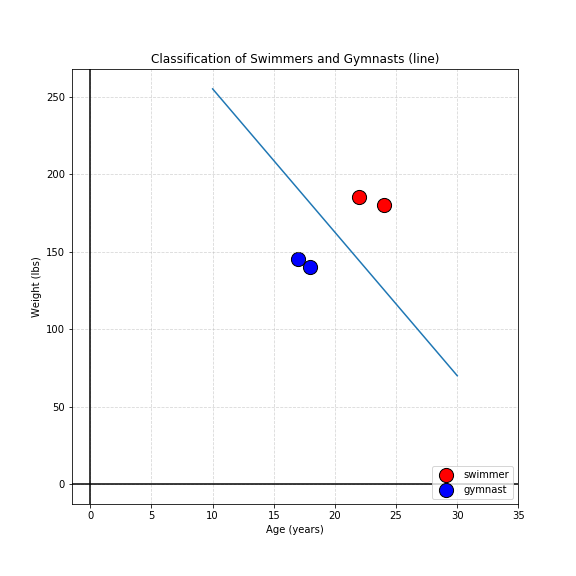
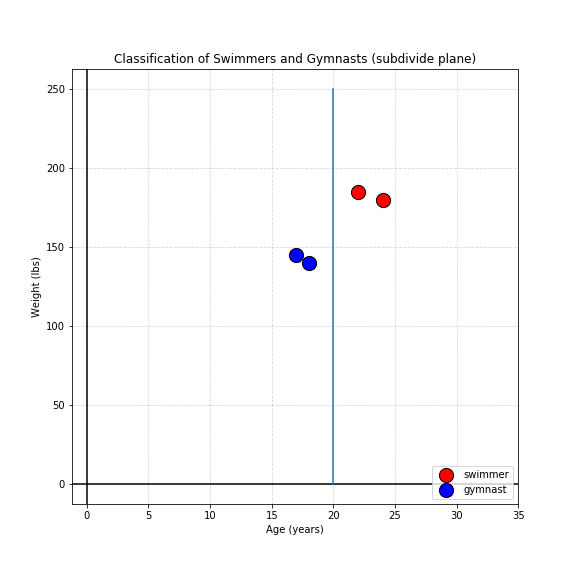
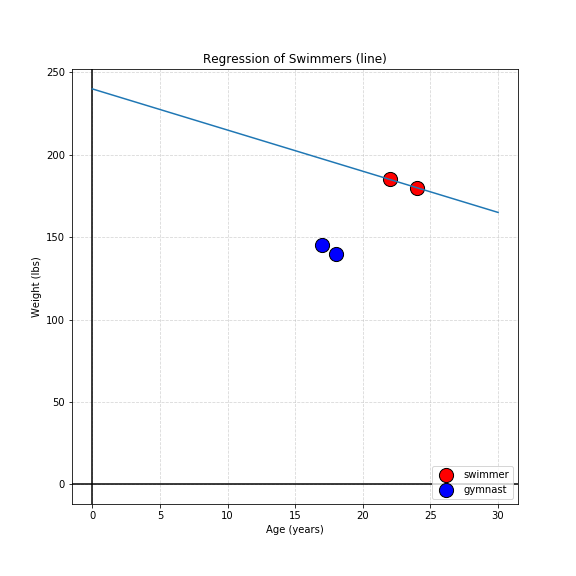
Unit 4-1 Vectors and Scalars

* Scalars
  + A scalar is a single number. Scalars are measurements with magnitude but not direction such as length, distance, and speed. A scalar is typically written as a non-bold, lowercase character.
    - A car is moving at 30.4 miles per hour. Its speed, sss, is a scalar because it's the single number 30.4. It we added direction to our quantity, it would not longer be a scalar.
    - The essay contains 37 words. The number of words, nnn, is a scalar because it's the single number 37.
    - I gained -5 lbs. The change in weight, www, is a scalar because it's the single number -5. (Because it's a negative number, it means five pounds were lost.)
  + In linear algebra, if you see a single, non-bold, lowercase character, it's safe to assume that it represents a scalar value.
* Vectors: Component Form
  + A vector is an ordered list of scalars. This is called the "component form."
  + In linear algebra, if you see a list of numbers enclosed in parentheses or brackets, you can assume it's a vector. A vector is typically depicted either as a bold, lowercase letter (e.g., **a** ) or a lowercase letter with an arrow on top .
  + For example, here are three vectors:
    - 
  + In this lesson, we'll denote vectors as bold, lowercase letters. Boldface is typically preferred in math texts. However, boldface isn't easy to draw by hand, so you may prefer the arrow notation when handwriting vectors.
* Scalars, Vectors, and Data Science
  + In most machine-learning algorithms, each data point must be converted to a vector. Why? Once all data points are vectors, they can be thought of as points in space. For example, if we have two groups of points in space, we can solve for a hyperplane that separates these points.
  + Let's suppose that a 24-year-old Olympic swimmer weighs 180 pounds and is 72 inches tall. Each of these attributes is a scalar that measures something. Together, they can be written as a vector:
  + **a**=(24,180,72)
* Vectorizing Data
  + Not all attributes are numeric, but we still have to convert them into scalars. The process of converting non-numeric data points to vectors is called vectorizing the data.
  + In scikit-learn, any function that includes the word vectorizer takes non-numeric input and returns numeric vectors as the output. CountVectorizer(), for example, converts a text document into a vector of word counts in which each component is how many times a particular word appears in the document.
  + Let's see how we can create a simple categorical vectorizer. Suppose each athlete from earlier is either a swimmer or a gymnast. This is a categorical (non-numeric) feature. So, we'll have to transform it to a scalar. Suppose 1 indicates swimmer and 0 indicates gymnast. Adding this scalar to our vector yields a new vector — now, the categorical information is numeric. If our athlete from earlier is a swimmer, the new vector will be:
  + **a**=(24,180,72,1)
  + By vectorizing an Olympic athlete, we're modeling a complex athlete as a sequence of four numbers. We're leaving out a great deal of information, but hopefully we've retained what's needed for prediction. Otherwise, we might have to think more about what other attributes are important to vectorize.
* Categorical Vectorizer
  + This technique of converting a categorical feature to one or more binary scalars (i.e., each having the value 0 or 1) is called creating dummy variables (or indicator variables).
  + Suppose we have a single categorical feature that has N possible categories. For regression models, we often prefer to convert this feature into N−1 separate scalars to vectorize it.
  + For example, suppose a "sport" feature falls into one of the categories: swimmer, gymnast, or fencer (N=3). To vectorize this, we'll create two scalars. The first will be 1 if a swimmer, 0 otherwise. The second will be 1 if a gymnast, 0 otherwise. If both are 0, then the athlete must be a fencer. So:
    - Swimmer will be vectorized as (1,0).
    - Gymnast will be vectorized as (0,1)
    - Fencer will be vectorized as (0,0)
  + If a vector only consists of binary scalars, it's called a binary vector.
* Categorical Vectorizer: Other Methods
  + You may be wondering why we might prefer to represent N categories using N−1 scalars instead of N scalars. Using N scalars — one binary scalar per category — is called a one hot encoding.
  + In the previous slide, notice how fencer was vectorized to (0,0) — all zeros. When we use N−1 scalars, having a vector of all zeros is meaningful. When we use N scalars, all zeros can never occur, as it doesn't correspond to a category.
  + Why is this important? In regression models (e.g., linear/logistic regression), the bias is the value where all variables are 0. So, if we use a one hot encoding, then the intercept is difficult to interpret.
  + Note: All types of data can be vectorized, including text, images, and sound. However, we often must leave out some information during vectorization. For example, it's difficult to store the relationship between every pair of words in a document numerically. Instead, we typically assume that nearby words are most important for understanding a given word. Words that are far away are typically ignored when vectorizing a particular word in a document (e.g., for sequence learning).
* Vectors: Coordinate Points
  + Visualizing vectors allows us to gain insight into how many algorithms work. Luckily, visualizing low-dimensional vectors is a straightforward process.
  + You might already know how to visualize an ordered list of numbers — for example, as a point on a graph. To plot the point (3,−2) on a coordinate grid, draw a point that is three units to the right of and two units down from the origin.
  + 
* Vectors: Visualization
  + Let's use a coordinate grid to understand some athlete data by first vectorizing each athlete. Suppose we know the age, weight, and sport (swimming = 1 or gymnastics = 0) of four Olympic athletes:
    - Swimmer: (24,180,1)
    - Swimmer: (22,185,1)
    - Gymnast: (18,140,0)
    - Gymnast: (17,145,0)
  + To better understand this data, we'll plot the age and weight as points on a coordinate grid, just as we discussed earlier. As you might notice, we end up with two clusters of points — one in the upper-right corner (swimmers) and one in the lower-left corner (gymnasts). Here, we're using different colors to denote the sports, but you could also plot the sport along a third dimension.
* Vectors: Classification
  + Notice that the two clusters can be separated by a line. (This is how logistic regression and support vector machines work.) The swimmers could also be classified using only vertical and horizontal lines, having an age older than 20 and/or a weight greater than 160. (This is how decision trees work.)
  + In the case of fitting an arbitrary boundary line, the purpose of linear algebra is to compute the equation of the line that best separates these two groups.
  + 
  + 
* Classification
  + When we say "compute the equation of the line," we mean to solve for two unknown coefficients, A and B (in these lessons, non-bold uppercase letters will denote coefficients, even though coefficients are scalars). These coefficients satisfy the equation:
  + weight=A \*age +B
  + In this case, any value above the line will be classified as a swimmer. Any value below the line will be classified as a gymnast.
  + How would we do this? In this case, we have a constraint. We want this line to be as "in between" the two data clusters as possible. So, we'll find the line where its points are farthest from both clusters. When we express this constraint mathematically, we get a score of how "good" our coefficients are. Using calculus, we can then solve for coefficients that maximize the value of the constraint.
* Parametric Models
  + The process we just illustrated is used for nearly all parametric machine-learning models. Parametric models rely on mathematical formulas with unknown coefficients/parameters, such as linear regression, logistic regression, and neural networks.
  + We can think of most parametric models using the following framework, as we must solve for the unknown coefficients/parameters:
  + 1) **Model**: An equation with coefficients for which to solve.
  + 2) **Constraint**: Used to know how "good" a set of coefficients is.
  + 3) **Optimization technique**: Finds ideal coefficients.
  + Linear regression is just the model (No. 1) —we can utilize different constraints (No. 2) and different optimization techniques (No. 3) that are independent of that model. Later in this unit, we’ll elaborate on this in more detail.
* Vectors: Regression
  + We saw that a line can be placed in between two groups of data. However, suppose we want to predict the weight of a swimmer given age (supposing this relationship even exists among Olympic swimmers). Instead of using a line to separate two groups, we'll draw the line through the swimmer data points.
  + This is called **regression**. Instead of using the line to discriminate between two groups and predict categories, we'll use it to help predict exact numbers.
  + 
  + For linear regression in this unit:
    - 1) **Model** is still y=Ax+B (or a general form we'll discuss later).
    - 2) **Constraint** is minimizing the mean squared distance of the actual data points from our model's predictions.
    - 3) **Optimization** technique is using algebra to solve for an exact solution. (For large systems, you might practically use an optimization technique such as gradient descent.)