Introduction to Classification and Regression Teaching Guide
Find My Code Here: https://github.com/kdutta9/CS-Education

Note: Ths guide isn't exhaustive, but it breaks down the overall session, pacing, and goals. Take a look at the code if you don't understand something, as I used open-source functions with great documentation! Your explanations should definitely be in more depth to a student. Hope this helps.

Setup:

- Preferably, Jupyter Notebook or any iPython notebook. However, any Python IDE will be fine.

Goal:

- Introduce students to classification and regression in machine learning.
- Give them the tools to execute individual projects.

Guide:

1. Explain classification to a student.
    a. Overview:
    Classification is just putting things into categories, based on some similarities. If we see a furry animal with four legs and a tail wearing a leash, we can categorize it as a dog. We know this from seeing dogs in our life. Imagine some alien trying to identify the animal - it wouldn't know how, because it's never seen it. But if we show it enough dogs, the alien will be able to predict if some random animal is a dog or not.
    Similarly, we want computers to categorize some object, based on its *features*.
    How do we do that?
2. The Code
    a. Open up *cancer-prediction-starter.ipynb.* We will try to predict if a patient has breast cancer.
    On the notebook, we have a guide:

        1. Collect data.
        2. Create a training set and testing set.
            a. A training set builds our model. Our testing set is what we use to evaluate our model's predictions.
        3. Choose a model, based on our goals and data.
            a. See ML models with sklearn here.
            b. Alternatively, you can build your own model to optimize to your data.
        4. Make predictions with our model.

5. Evaluate the model by comparing predictions to our testing set.

Our data is given to us, by the breast cancer data. Try printing it out and seeing what it contains.
When we create a model, we need to make a training and testing set, in order to use the training set to make predictions and use the testing set to see if our predictions are correct. Now, we need to actually create a training and testing set. Luckily, we have a train_test_split function that could work nicely.
We have our data! Let's try using it so our code can see all of it and make predictions based on the input. We use *LogisticRegression*, because it is most optimal for binary classification, which is what we're trying to classify in this case (i.e. has or does not have breast cancer).
Now, we have developed a *model*. Let's see how we did. Make predictions using the *predict* method. Evaluate using the many methods at our disposal: *confusion_matrix, classification_report, mean_squared_error.*

3. Regression: This lesson should break down the same way, but we use *housing-prices-starter.ipynb* and the solutions are in *housing-prices.ipynb.* Instead of classifying based on features, we try to create a line based on the numerical inputs. Basically like $y = mx + b$, but more like $y = m_1x_1 + m_2x_2 + m_3x_3 + ... + b$