SEND FEEDBACK

Boston Housing Example

SageMaker Sessions & Execution Roles

SageMaker has some unique objects and terminology that will become more familiar over time. There are a few objects that you'll see come up, over and over again:

- **Session** - A session is a special *object* that allows you to do things like manage data in S3 and create and train any machine learning models; you can read more about the functions that can be called on a session, at this documentation. The `upload_data` function should be close to the top of the list! You'll also see functions like `train`, `tune`, and `create_model` all of which we'll go over in more detail, later.
- **Role** - Sometimes called the *execution role*, this is the IAM role that you created when you created your notebook instance. The role basically defines how data that your notebook uses/creates will be stored. You can even try printing out the role with `print(role)` to see the details of this creation.

Uploading to an S3 Bucket

Another SageMaker detail that is new is the method of data storage. In these instances, we'll be using S3 buckets for data storage.

> *S3 is a virtual storage solution that is mostly meant for data to be written to few times and read from many times. This is, in some sense, the main workhorse for data storage and transfer when using Amazon services. These are similar to file folders that contain data and metadata about that data, such as the data size, date of upload, author, and so on.*

S3 stands for Simple Storage Service (S3).

After you upload data to a session, you should see that an S3 bucket is created, as indicated by an output like the following:

```
INFO: sagemaker: Created S3 bucket: <message specific
```

If you'd like to learn more about how we're creating a csv file, you can check out the pandas documentation. Above, we are just concatenating x and y data sets as columns of data (`axis=1`) and converting that pandas dataframe into a csv file using `.to_csv`.

Boston Housing Data

For our very first time using SageMaker we will be looking at the problem of estimating the median cost of a house in the Boston area using the Boston Housing Dataset.

We will be using this dataset often throughout this module as it provides a great example on which to try out all of SageMaker's features.

In addition, we will be using a random tree model. In particular, we will be using the XGBoost algorithm. The details of XGBoost are beyond the scope of this module as we are interested in learning about SageMaker. If you would like to learn more about XGBoost I would recommend starting with the documentation which you can find at https://xgboost.readthedocs.io/en/latest/

The notebook we will be working though in this video and in the following two videos can be found in the `Tutorial` directory and is called `Boston Housing - XGBoost (Batch Transform) - High Level.ipynb`. Now that you know why **Boston Housing** and **XGBoost** are in the name, let's talk a bit about the rest of it.

First, **Batch Transform** is the method we will be using to test our model once we have trained it. This is something that we will discuss a little more later on.

Second, **High Level** describes the API we will be using to get SageMaker to perform various machine learning tasks. In particular, it refers to the Python SDK whose documentation can be found here: https://sagemaker.readthedocs.io/en/latest/. This high level approach simplifies a lot of the details when working with SageMaker and can be very useful.