**Updating a Model**

In this lesson we are going to take a look at updating an existing endpoint so that it conforms to a different endpoint configuration. There are many reasons for wanting to do this, the two that we will look at are, performing an A/B test and updating a model which is no longer performing as well.

To start, we will look at performing an A/B test between two different models. Then, once we've decided on a model to use, updating the existing endpoint so that it only sends data to a single model.

For the second example, it may be the case that once we've built a model and begun using it, the assumptions on which our model is built begin to change.

For instance, in the sentiment analysis examples that we've looked at our models are based on a vocabulary consisting of the 5000 most frequently appearing words in the training set. But what happens if, over time, the usage of words changes? Then our model may not be as accurate.

When this happens we may need to modify our model, often this means re-training it. When we do, we'd like to update the existing endpoint without having to shut it down. Fortunately, SageMaker allows us to do this in a straightforward way.

NEXT

# Building a Sentiment Analysis Model

To begin with we will create an XGBoost model similar to the ones that we have constructed in the past in order to predict the median housing cost in Boston.

The difference this time is that we are using a hybrid approach, including both the high level and low level functionality. In this case we use the high level approach to train a model (to produce model artifacts) and then we use the low level approach to construct the model itself and to construct the endpoint configuration. The reason for this is so that we can have more control over how our endpoint behaves.

# Building a Sentiment Analysis Model

Depending on the application you have in mind for a particular machine learning model, accuracy may not always be the metric you wish to optimize. There may be some other constraints on getting the model to work in production. For example, your model may not be very easy to interpret or maybe performing inference for a particular model may be too costly.

In any case you may want to try alternative models. In the example we are working on here we construct a linear learner model as an alternative to the previously created XGBoost model.

**Note:** It is important to notice that the result returned by the linear learner model is json, compared to the csv data returned by the XGBoost model. You can't always assume that different models will return data in the same way although typically the return type is specified in the documentation.

# Combining the Models

Using the low level approach to creating endpoint configurations allows us to create endpoints that are more sophisticated. For example, endpoints which receive data and route that data to one of many different models. In the example here we are only using two different models but there may be situations in which you would want more.

In addition, SageMaker provides functionality to update an existing endpoint so that it conforms to a different endpoint configuration. Further, SageMaker does this in a way that does **not** require the existing endpoint to be shut down.

This is very beneficial as you may be working in an environment where there are other services that depend on your deployed endpoint.

# Mini-Project: Updating a Sentiment Analysis Model

In this mini-project we will take a look at situation in which we have a trained model which is working well, but then something changes with the underlying distribution on which our model is based. First we need to take a look at what might be the problem. Then we want to create a new, updated model and replace our old model without taking down the corresponding endpoint.

This mini-project notebook is called IMDB Sentiment Analysis - XGBoost (Updating a Model).ipynb and can be found inside of the Mini-Projects folder.

# SageMaker Retrospective

In this module we looked at various features offered by Amazon's SageMaker service. These features include the following.

* **Notebook Instances** provide a convenient place to process and explore data in addition to making it very easy to interact with the rest of SageMaker's features.
* **Training Jobs** allow us to create model artifacts by fitting various machine learning models to data.
* **Hyperparameter Tuning** allow us to create multiple training jobs each with different hyperparameters in order to find the hyperparameters that work best for a given problem.
* **Models** are essentially a combination of model artifacts formed during a training job and an associated docker container (code) that is used to perform inference.
* **Endpoint Configurations** act as blueprints for endpoints. They describe what sort of resources should be used when an endpoint is constructed along with which models should be used and, if multiple models are to be used, how the incoming data should be split up among the various models.
* **Endpoints** are the actual HTTP URLs that are created by SageMaker and which have properties specified by their associated endpoint configurations. **Have you shut down your endpoints?**
* **Batch Transform** is the method by which you can perform inference on a whole bunch of data at once. In contrast, setting up an endpoint allows you to perform inference on small amounts of data by sending it do the endpoint bit by bit.

In addition to the features provided by SageMaker we used three other Amazon services.

In particular, we used **S3** as a central repository in which to store our data. This included test / training / validation data as well as model artifacts that we created during training.

We also looked at how we could combine a deployed SageMaker endpoint with **Lambda** and **API Gateway** to create our own simple web app.

NEXT

# Cleaning Up Your AWS Account

Once you have finished making use of Amazon's services you should make sure to clean up your account. One of the main reasons for this is so that you don't get an unexpected bill!

### **Have you cleaned up?**

Task List

* 

Shut down any notebook instances that are running.

* 

Shut down any endpoints that are running.

* 

Clean up your S3 buckets, provided you are finished with them. **Note:** Deleting an S3 bucket can not be undone so only do this if you are completely finished.

* 

Delete any unused Lambda functions.

* 

Remove any unused APIs created using API Gateway

NEXT

# Tips and Tricks

# SageMaker Documentation

* **Developer Documentation** can be found here: <https://docs.aws.amazon.com/sagemaker/latest/dg/>
* **Python SDK Documentation** (also known as the high level approach) can be found here: <https://sagemaker.readthedocs.io/en/latest/>
* **Python SDK Code** can be found on github here: <https://github.com/aws/sagemaker-python-sdk>