## Lab 3-02: Introducing Jupyter Notebooks (AWS SageMaker)

### Lab Prerequisites

* Familiarity with basic AWS Cloud Computing concepts and terminology
* An AWS account with an active subscription

### Service Introduction

You may create and share documents with live code, mathematics, visuals, and narrative text using the open-source web application Jupyter Notebook.

You can build and operate Jupyter notebooks from your DLAMI instance using a Jupyter notebook server. You can use Jupyter notebooks to run Machine Learning (ML) experiments for inference and training while utilizing the AWS infrastructure and DLAMI package libraries.

### Case Study Enterprise EMRs – [Savana](https://savanamed.com/)

Background

Savana offers processing and analytic services for Electronic Medical Records (EMRs). With more than 180 hospitals spread over 15 nations, Savana, a Madrid-based company, runs one of the world's largest Artificial Intelligence (AI), multicentric, and multilingual research networks.

Savana, a company, established in Madrid, assists healthcare providers in maximizing the value of their EMRs for research. To produce pertinent results for healthcare and life science providers looking into disease prediction and therapy, it blends research-grade methods with Natural Language Processing (NLP) and predictive analytics. It can process the massive amounts of data needed to run machine learning algorithms using Amazon Web Services (AWS). Along with meeting regional regulatory requirements, it can scale its infrastructure to support its rapid growth across international markets, including the US, Latin America, and Western Europe. Compared to the prior technology, it has shortened study processing times by 25% and IT expenditures by up to 90%.

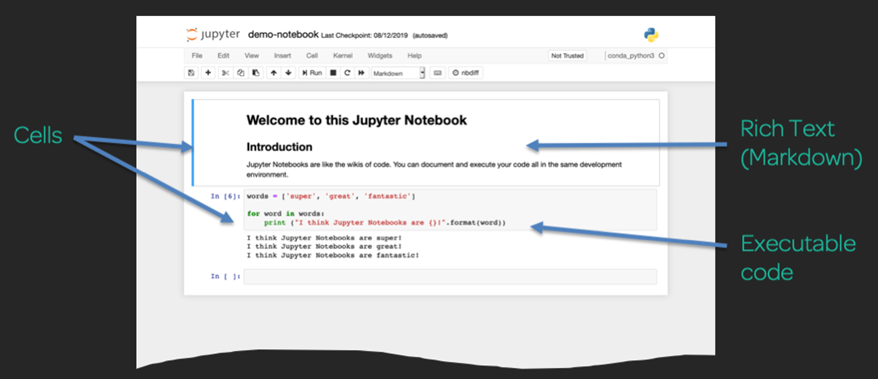
Challenge

You are working in an organization. You are provided some sample heights and weights of individual penguins. You have to use the data (stored in a csv file) to train a data model and create inferences so that you can guess what a penguin weighs if you know its height. Hence, you need to build a machine learning model that tells you approximately what a penguin should weigh based on height.

Proposed Solution

The solution is for you to use AWS services to automate all the tasks. You create an AWS SageMaker hyperparameter tuning job with different ranges of values for the hyperparameter to find the best configuration, which minimizes the validation: objective\_loss metric. The reason that you use this metric is that it is used in multi-classification problems. This metric measures the performance of the classification model, and what it does is repeatably calculate the difference between the values that the model predicts and the actual values of a label. Hence, every time it passes over data and makes predictions, it recalculates the objective loss and tries to minimize this value overall. Hence, that is the task here. AWS recommends that you minimize this value when using it as our objective metric.

Lab Diagram



*Figure 6-50: Lab Diagram*

Implementation Steps

1. Navigate to Jupyter Notebook.
2. Browse Jupyter Notebooks.
3. Create New Folders and Files
4. Build Your Own Notebook
5. Use Markdown to Add Richly Formatted Text to a Notebook

### Solution

|  |
| --- |
| **Step 1: Navigate to Jupyter Notebook**   1. Login to **AWS Management Console**.      1. Search for **SageMaker** in the search bar.      1. Click on **Notebook Instances** under the dashboard.   **Step 2: Browse Jupyter Notebooks**   1. Click on **Open Jupyter**.     **Step 3: Create New Folders and Files**   1. Click on **New**. 2. Click on **Folder**.      1. Select Folder. 2. Click on **Rename**.      1. Define Name. 2. Click on **Rename**.      1. Click on the **new folder**.      1. Select the item to perform the action.      1. Go through to the **Introduction to Jupyter Notebooks**.      1. Click on **Insert.** 2. Click on **Insert Cell Below.**      1. Select the cell. Click on **Run.**      1. Go through the output result.      1. Select the cell. Click on **Run**.      1. Select the cell. Click on **Run**.      1. Go through the output result.      1. Select the cell. Click on **Run**.      1. Go through the output result.      1. Select the cell. Click on **Run**.      1. Go through the output result.      1. Similarly, select each cell and click Run to see the output.      1. Select the cell. Click on **Run**.      1. Go through the output result.      1. Select the cell. Click on **Run.**      1. Go through the output result.      1. Go through penguin-data.csv        1. Select the cell. Click on **Run.**      1. Go through the output result.      1. Select the cell. Click on **Run.**      1. Go through the output result.      1. Select the cell. Click on **Run.**      1. Go through the output result.      1. Select the cell. Click on **Run.**      1. Go through the output result.      1. Select the cell. Click on **Run.**      1. Go through the output result.      1. Select the cell. Click on **Run.**      1. Go through the output result.     **Step 4: Build Your Own Notebook**   1. Click on **conda\_python3.**      1. Click on **Untitled**.      1. Define notebook name. 2. Click on **Rename**.     **Step 5: Use Markdown to Add Richly Formatted Text to a Notebook**   1. Click on **Markdown.**      1. Add **#Welcome to my notebook** in the cell.      1. Click on **File**. 2. Click on **Download as**. 3. Click on **Notebook (.ipynb).** |