# Hands-on Lab: Basic Model Training with Amazon SageMaker

This lab will guide you through building and training a machine learning model using Amazon SageMaker. We’ll focus on a simple classification problem to demonstrate the core concepts of SageMaker, including environment setup, data preparation, model training, and result analysis.

## Prerequisites

* An AWS account with SageMaker access
* Basic understanding of Python and machine learning concepts
* Familiarity with Jupyter notebooks

## Lab Steps

### 1. Environment Setup

1. Open the Amazon SageMaker console at https://console.aws.amazon.com/sagemaker/
2. Create a new SageMaker notebook instance:
   * Choose “Notebook instances” from the left navigation pane
   * Click “Create notebook instance”
   * Enter a name for your notebook instance
   * Choose an instance type (e.g., ml.t3.medium)
   * Select “Create a new role” under IAM role
   * Choose “Create role”
   * Click “Create notebook instance”
3. Once the instance status is “InService”, click “Open Jupyter” to launch JupyterLab.

### 2. Data Preparation

1. In JupyterLab, create a new Jupyter notebook.
2. Install and import necessary libraries:

!pip install pandas numpy scikit-learn  
import pandas as pd  
import numpy as np  
from sklearn.model\_selection import train\_test\_split  
from sklearn.datasets import load\_iris

1. Load and prepare the Iris dataset:

iris = load\_iris()  
X, y = iris.data, iris.target  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
train\_data = pd.DataFrame(X\_train, columns=iris.feature\_names)  
train\_data['target'] = y\_train  
test\_data = pd.DataFrame(X\_test, columns=iris.feature\_names)  
test\_data['target'] = y\_test  
  
train\_data.to\_csv('train.csv', index=False, header=False)  
test\_data.to\_csv('test.csv', index=False, header=False)

### 3. SageMaker Setup

Set up SageMaker session and role:

import sagemaker  
from sagemaker import get\_execution\_role  
  
role = get\_execution\_role()  
session = sagemaker.Session()  
  
bucket = session.default\_bucket()  
prefix = 'sagemaker/DEMO-iris'  
  
train\_location = session.upload\_data('train.csv', bucket=bucket, key\_prefix=prefix)  
test\_location = session.upload\_data('test.csv', bucket=bucket, key\_prefix=prefix)

### 4. Model Training

Configure the XGBoost algorithm and start the training job:

from sagemaker.amazon.amazon\_estimator import get\_image\_uri  
  
container = get\_image\_uri(session.boto\_region\_name, 'xgboost', '1.0-1')  
  
xgb = sagemaker.estimator.Estimator(container,  
 role,   
 instance\_count=1,   
 instance\_type='ml.m4.xlarge',  
 output\_path=f's3://{bucket}/{prefix}/output',  
 sagemaker\_session=session)  
  
xgb.set\_hyperparameters(max\_depth=5,  
 eta=0.2,  
 gamma=4,  
 min\_child\_weight=6,  
 subsample=0.8,  
 objective='multi:softprob',  
 num\_class=3,  
 num\_round=100)  
  
xgb.fit({'train': train\_location, 'validation': test\_location})

### 5. Model Deployment

Deploy the trained model and make predictions:

xgb\_predictor = xgb.deploy(initial\_instance\_count=1, instance\_type='ml.m4.xlarge')  
  
from sagemaker.predictor import csv\_serializer, json\_deserializer  
  
xgb\_predictor.content\_type = 'text/csv'  
xgb\_predictor.serializer = csv\_serializer  
xgb\_predictor.deserializer = json\_deserializer  
  
result = xgb\_predictor.predict(test\_data.iloc[:, :-1].values).decode('utf-8')  
predictions = np.fromstring(result[1:-1], sep=',')  
predictions = predictions.reshape(-1, 3)

### 6. Evaluate Results

Calculate accuracy:

predicted\_labels = np.argmax(predictions, axis=1)  
accuracy = (predicted\_labels == y\_test).mean()  
print(f"Model accuracy: {accuracy:.2f}")

### 7. Clean Up Resources

Delete the endpoint:

* Go to the SageMaker console
* Choose “Inference” > “Endpoints”
* Select your endpoint
* Choose “Actions” > “Delete”

Delete the notebook instance:

* Go to the SageMaker console
* Select “Notebook instances” from the left navigation pane
* Select your notebook instance
* Choose “Actions” > “Stop”
* Once stopped, choose “Actions” > “Delete”

Delete other resources:

* Delete the S3 bucket created for this lab
* Delete any CloudWatch logs created during the lab

## Common Mistakes and Best Practices

* Ensure your data is in the correct format for the chosen algorithm.
* Always clean up resources after use to avoid unnecessary charges.
* Consider using SageMaker’s hyperparameter tuning capabilities for better results.
* Use SageMaker Experiments to track different versions of your model and their performance.
* Ensure your S3 buckets and IAM roles have appropriate permissions set.

By completing this lab, you’ve gained hands-on experience with the core components of Amazon SageMaker, including data preparation, model training, deployment, and evaluation.