

# Cloud Computing and Big Data - Spring 2020

## Homework Assignment 3

### Assignment:

In this assignment you will Implement a machine learning model to predict whether a message is spam or not. Furthermore, you will create a system that upon receipt of an email message, it will automatically flag it as spam or not, based on the prediction obtained from the machine learning model.

### Outline:

This assignment has the following components:

#### 1. Complete tutorial for using Amazon SageMaker on AWS.

- a. Follow the following AWS tutorial on how to use Amazon SageMaker to implement the required model<sup>1</sup>:  
<https://aws.amazon.com/getting-started/hands-on/build-train-deploy-machine-learning-model-sagemaker/>
- b. The purpose of the tutorial is to familiarize you with Amazon Sagemaker and the basic components of SageMaker.

#### 2. Implement a Machine Learning model for predicting whether an SMS message is spam or not.

- a. Follow the following AWS tutorial on how to build and train a spam filter machine learning model using Amazon SageMaker:  
<https://github.com/aws-samples/reinvent2018-srv404-lambda-sagemaker/blob/master/training/README.md>
- b. The resulting model should perform well on emails as well, which is what the rest of the assignment will focus on.
- c. Deploy the resulting model to an endpoint (E1).

#### 3. Implement an automatic spam tagging system.

- a. Create an S3 bucket (S1) that will store email files.
- b. Using SES, set up an email address, that upon receipt of an email it stores it in S3.

---

<sup>1</sup> <https://aws.amazon.com/sagemaker>

- i. Confirm that the workflow is working by sending an email to that email address and seeing if the email information ends up in S3.
- c. For any new email file that is stored in S3, trigger a Lambda function (LF1) that extracts the body of the email and uses the prediction endpoint (E1) to predict if the email is spam or not.
  - i. You might want to strip out new line characters “\n” in the email body, to match the data format in the SMS dataset that the ML model was trained on.
- d. Reply to the sender of the email (it could be your email, the TA’s etc.) with a message as follows:

*“We received your email sent at [EMAIL\_RECEIVE\_DATE] with the subject [EMAIL\_SUBJECT].*

*Here is a 240 character sample of the email body:  
[EMAIL\_BODY]*

*The email was categorized as [CLASSIFICATION] with a [CLASSIFICATION\_CONFIDENCE\_SCORE]% confidence.”*

- i. Replace each variable “[VAR]” with the corresponding value from the email and the prediction.
  - ii. The purpose of this step is to facilitate easy testing.

#### **4. Create an AWS CloudFormation template for the automatic spam tagging system.**

- a. Create a CloudFormation template (T1) to represent all the infrastructure resources (ex. Lambda, SES configuration, etc.) and permissions (IAM policies, roles, etc.).
- b. The template (T1) should take the prediction endpoint (E1) as a stack parameter.

**Acceptance criteria:**

1. TAs should be able to email the unique email address submitted as part of the assignment and they should be able to get reasonable predictions (spam/not spam) for the emails they send.
2. TAs should be able to stand up the CloudFormation template (T1) within a separate account, using their own prediction endpoint (E1'), and successfully test the system.
  - a. This also assumes that you provide the TAs with the code for the Lambda function (LF1).

## ANNEX

### Architecture Diagram

