

# AWS\_Framework

February 28, 2019

## 1 AWS - Framework

### 1.1 Description

This workbook puts together all components needed to set up the AWS network for the VWT Machine Learning Framework.

### 1.2 Content

Below are the stages of the network in which set-up will be detailed throughout the workbook:

- SageMaker
- API Gateway
- Lambda Functions
- Endpoint
- Step Functions
- Cloudwatch

### 1.3 Diagram

### 1.4 ML System

#### 1.4.1 SageMaker

Diagram

SageMaker is the networks main place for training the machine learning models. For the purpose of this workbook SageMaker will act as the host of all the services in which AWS will use to perform all machine learning.

SageMaker will be used to re-train the various machine learning models that the VoX Wallet will provide to the user or operator. Each of these models will have different datasets stored in one or multiple S3 buckets and will use seperate algorithms for the training.

Data labelling will need to take place on these datasets for any supervised learning models to be run. Depending on the size of the data Auto Data Labelling can take place (usually at least 5,000 datapoints). Datasets with smaller size will need Human labelling, AWS provides Labelling Jobs which are sent to workers within the AWS network. The difficulty then comes in how these workers, or a machine will be able to determine the labels of the data, for example guessing which of the datapoints is a problem gambler. Perhaps the data will need to be labeled by an operator,

using self exclusion as an anchor, as they will better understand which of their users are problem gamblers. Currently researching the potential of unsupervised machine learning algorithms, which therefore will not require data labelling.

Below is a list of other services that are involved in the machine learning processes and are managed by SageMaker:

- S3
- ECR
- Auto Scaling Instances

**S3** S3 is the storage service that AWS provides for any data or object. Each separate machine learning model in the ML System will require an S3 Bucket for both the data and the machine learning model. There will be a total of 5 S3 buckets required:

#### **S3 Data Buckets**

- S3 (Before Initial Deposit) Data Bucket
- S3 (Gambling Behaviour) Data Bucket

#### **S3 Model Buckets**

- S3 (Before Initial Deposit Problem Gambling) Model Bucket
- S3 (Gambling Behaviour Problem Gambling) Model Bucket
- S3 (Gambling Behaviour Marketing) Model Bucket

**ECR** ECR will hold the initial model it will hold the uploaded docker image.

**Auto Scaling Instances** This is the compute resource that will be used for the processing of the machine learning training. These instances will scale up and down depending on the amount of training data that is available.

## **1.5 Prediction Pipeline**

Diagram

### **1.5.1 API Gateway**

The API Gateway is an AWS service that enables you to create, publish, maintain, monitor, and secure your own REST and WebSocket APIs at any scale. You can create robust, secure, and scalable APIs that access AWS, as well as data stored in the AWS Cloud. This will enable data calls to be made straight from the user device to the cloud, the first step in order to obtain a ML prediction. It also provides support for different DNS calls, such as mobile and web.

To create an API Gateway go to the service in the AWS Management Console and click Create API. Once the API is created, create a new resource within the API Gateway. You can now add methods to this new resource. One of these methods will be to integrate with the Lambda Functions that are defined below in order to preprocess the incoming data.

API throttling allows you to set permissions as to whether certain API calls are valid or not. it has the following use cases:

- Deliver consistent applications by making sure that a single client is not suffocating your applications. Enhanced performance will drastically improve the end-user experience.
- Control user authentication and access by rate limiting APIs at various levels—resource, API or application.
- Design a robust API that can be leveraged by multiple groups based on their access level. Simplified API monitoring and maintenance can help reduce your costs.

Further details:

<https://medium.com/@asankha/exposing-aws-lambda-functions-with-the-api-gateway-fbdc15c19dce>

Once the data is preprocessed we can then invoke the machine learning model from the SageMaker endpoint.

<https://aws.amazon.com/blogs/machine-learning/call-an-amazon-sagemaker-model-endpoint-using-amazon-api-gateway-and-aws-lambda/>

### 1.5.2 Lambda Functions

The Lambda functions will connect to the API Gateway and act as a preparation filter to then be labeled at the SageMaker endpoint. The data will be distributed for the respective pre-processing algorithm, then computed using serverless.

Lambda function for predicting using the model.

**Filter Lambda Function Code** We may need to filter different data received in the API to separate lambda functions for preprocessing, as each dataset has a separate preprocessing method.

### 1.5.3 SageMaker Endpoint

After you deploy a model into production using Amazon SageMaker hosting services, your client applications use this API to get inferences from the model hosted at the specified endpoint. After the data has been preprocessed we then use the endpoint to invoke the model stored in the S3 Model Bucket and then make a prediction.

## 1.6 Monitoring & Updating

Diagram

### 1.6.1 Step Functions

The step functions will provide the commands for invoking of re-training for the SageMaker machine learning. These functions consist of JSON code that connects the triggers time based events from CloudWatch to all the services within SageMaker.

### Main State Machine

### 1.6.2 CloudWatch

CloudWatch provides monitoring across the ML System. It provides the ability to trigger events based on custom metrics that the administrator wishes to flag. The admin will be able to check how the system is running in order to fix or update certain services.

One of the requirements of the system is a time based event to be triggered by CloudWatch for re-training of new data over certain time horizons to be determined. This will stimulate the Step Functions listed above, and can be lowered or elongated depending on the requirements of accuracy improvement.

## 1.7 After AWS

Once the prediction is complete the result will be used differently depending on what was being predicted. For example, the prediction of a risk group for a user corresponding to problem gambling will be used differently to the prediction of how much a user will bet on a certain day.

Below is a list of the different places a prediction will be sent to:

- Intervention Model
- Operator

### 1.7.1 Intervention Model

The Intervention Model is a system that will use risk groups to put in place certain restrictions or other measures of prevention to the user. There will be a positive correlation between the risk score and the size of the restrictions that are enforced.

The intervention model may just be located on the blockchain, therefore all the process would need to do is to sent the ouput from the prediction model to this.

### 1.7.2 Operator

As part of the ML System Operators may recieve the predictions that are made on the data of users. An example of this may be where the machine learning model is predicting detailed insights of users that can be used for better marketing systems.

Again the output of the prediction model may just need to be stored or sent to the operator.

In [ ]: