Report 2: Student Progress Report

Please complete all fields.

Name:	**Rahul Ethiraj**
Company:	A9.com
Department/Team:	Advertising Technology – Newton team
Project Name:	Amazon Experimentation Framework & metrics- AXF

1. Brief description of the project:

- The central idea of the project is to prototype processing raw AAX logs in EMR using Spark. Experimentation notably controlled A/B testing is one of the primary mechanisms that enables measuring and introducing step changes aiming to improve the performance of Multi-Channel Advertising (MCA).
- Experiments introduce the change (treatment) in a controlled setting isolated from existing behavior (Control) and measure the performance difference between the existing behavior and the changed behavior.
- I plan to use statistical measures to check the significance of Treatment and Control on one another across different existing metrics in AAX console.
- The goal is to populate metrics report for every hourly log generated by AAX server within one hour and make PROD level old Ad Runway Metrics pipeline that also calculates statistical metrics such as p-value and confidence intervals.

2. Work done and progress made since Report 1:

- WebLab(sister team of ours) currently has the statistical metrics to check the significance of Treatment and Control over varied metrics. I am incorporating the same into the AXF console for its own specific metrics.
- I parsed logs varying from bid, impression, click, punt, nobid and aggregated the required metric results.
- Benchmarked existing DASK pipeline against the one I built using Spark and the results were comparable.
- Researched about other alternatives such as Kinesis, logstash, AdRunway thin, Elastic Search, Log Analyzer class, Timber, Athena, Prometheus, Netflix-Atlas and selected the framework that is more suited for my project requirements.
- Below are the log types along with source groups that I have parsed so far, and planning to do a couple of more before pushing the code to the production environment:
- Log types:
    - Bid - adx, mobile_rtb, rmx, rtb_other, video_rtb
    - Impression - adx, mobile_rtb, other, rmx, rtb_other, video_rtb
    - Click - adx, mobile_rtb, other, rmx, rtb_other, video_rtb
    - Punt - adx, mobile_rtb, other, rmx, rtb_other, video_rtb

- Ran a spark job to create an RDD from raw JSON AAX logs and generate AXF metrics, store aggregate metrics back in s3.
- Tested on different EMR configuration (memory/compute intensive hosts)
- Prototyped launching EMR cluster /running EMR steps on top of the cluster / terminate the cluster with Aegis. This involves creating a prod Apollo env hosting all the spark/hive jobs and launching directing from there.
- Test/Deploy environments to beta/gamma/prod /stabilize every production job.

3. Problems encountered:

- Fortunately, I don't have any problems relating to management, team or people.
- Due to data confidentiality, I have listed a few of code & environment-specific problems below that I have encountered at a very high level and how I have fixed them:

    - Problem Description:
        - Prototyped integrating automation steps with AAX-console Jupyter notebook. The goal is to have a notebook which can do all the 3 steps for us - create a cluster, run job, terminate cluster when done.
        - Solution:
            - Was encountering an issue with dependencies integrating EMR(Elastic-Map-Reduce) and Aegis (homegrown framework for automation of cluster jobs) framework, solved it later by putting version filter instance.
    - Problem Description:
        - Researched on how can we easily surface the monitoring data on Jupyter notebook and integrate with Jupyter notebook. The goal is for the user to get quick access to the EMR console/s3 bucket having job run logs for the EMR job-id.
        - Solution:
            - I solved this problem by surfacing an API via which we can retrieve the cluster master node IP address so that user can SSH to the node if needed for any debugging.
    - Problem Description:
        - Extended existing spark jobs to integrate with PMET (internal job monitoring tool).
        - Solution:
            - I made Aegis support this so that we can put XMONs on top of job stats.
    - Problem Description:
        - The requirement was to launch multiple jobs on the same cluster.
        - Solution:
            - Found a workaround by giving the same cluster name in job step so we can save time on cluster startup and cluster termination for every hour.
    - Problem Description:
        - Since read/write time from/to S3 is huge with Spark on EMR, my mentor recommended having following –
        - Solution:
            - One master job which generates & write intermediate data set required by multiple applications in S3 & each app computes aggregates based on app logic.