Richard Schaefer

Bike Sharing Project

AI Camp AWS ML Full Stack

Model Testing and Tuning on AWS (companion document to presentation and Colab notebook)

3 Models were tested on AWS

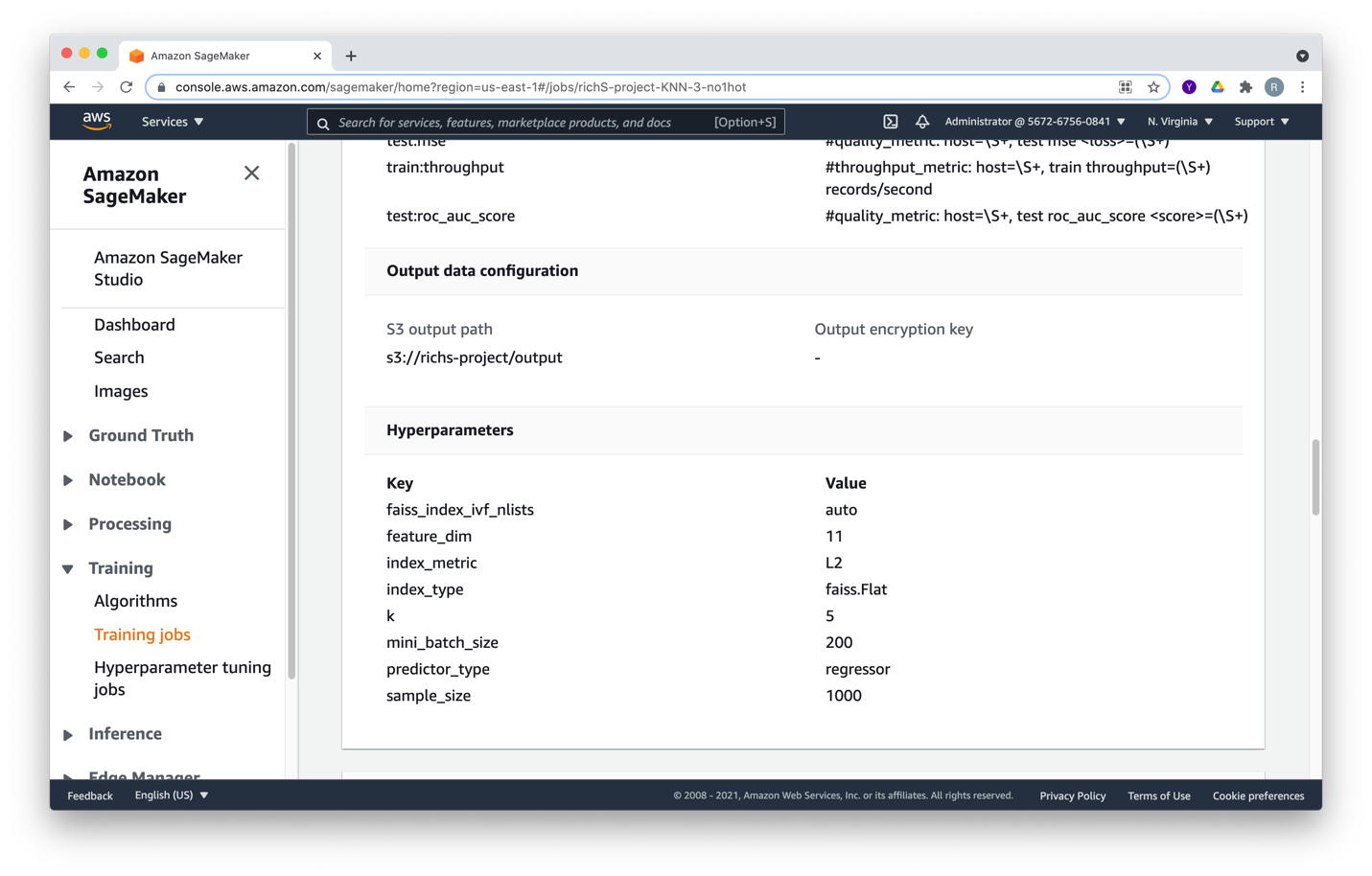
* KNN (built-in)
* XG Boost (built-in)
* Random Forest (custom)

KNN

Base-line runs of KNN were done with the dataset with and without one-hot-encoding.

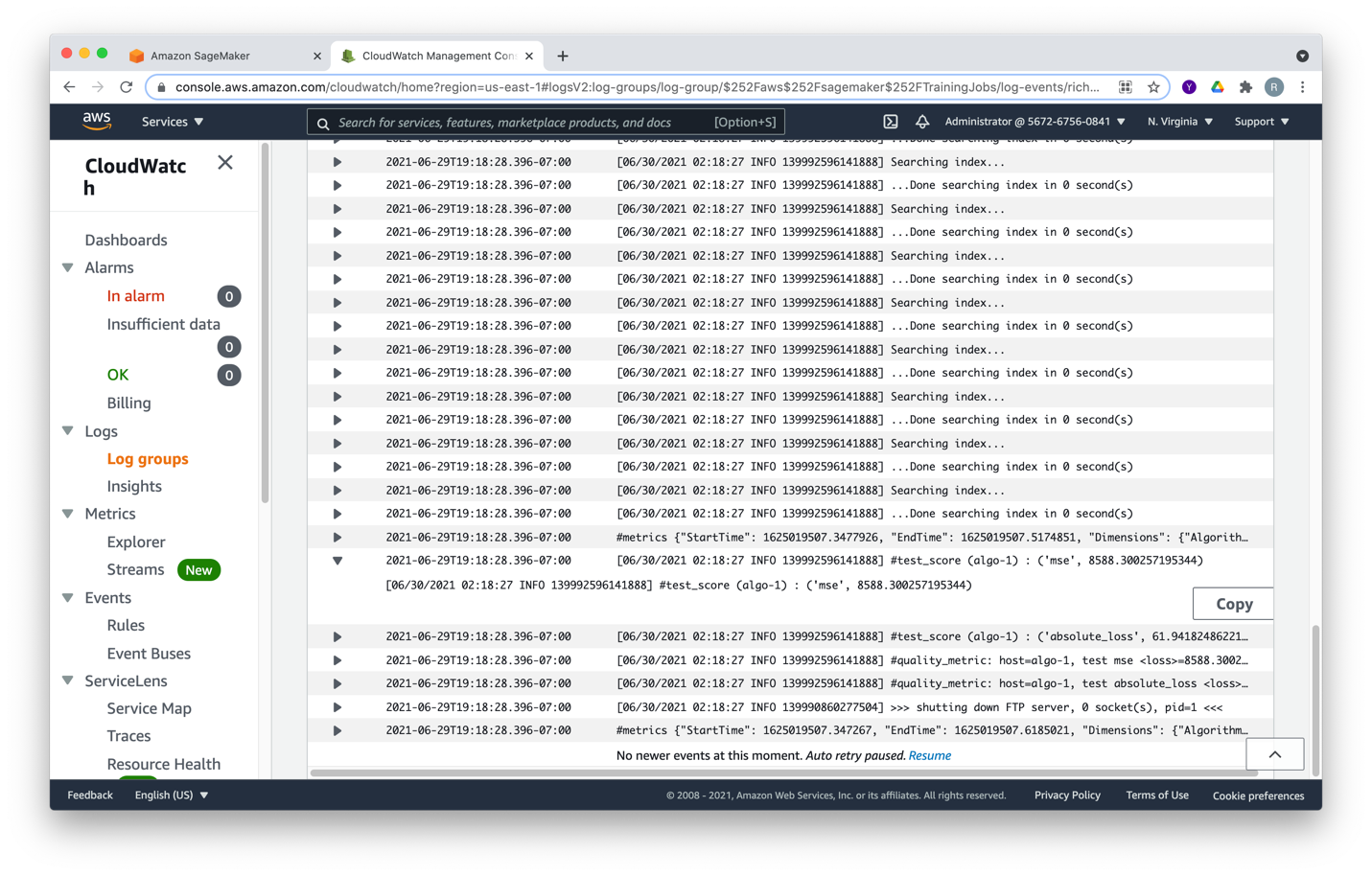
Base-line for run without 1-hot-encoding used the following hyperparameters:

K = 5, sample size = 1000, min-batch = 200

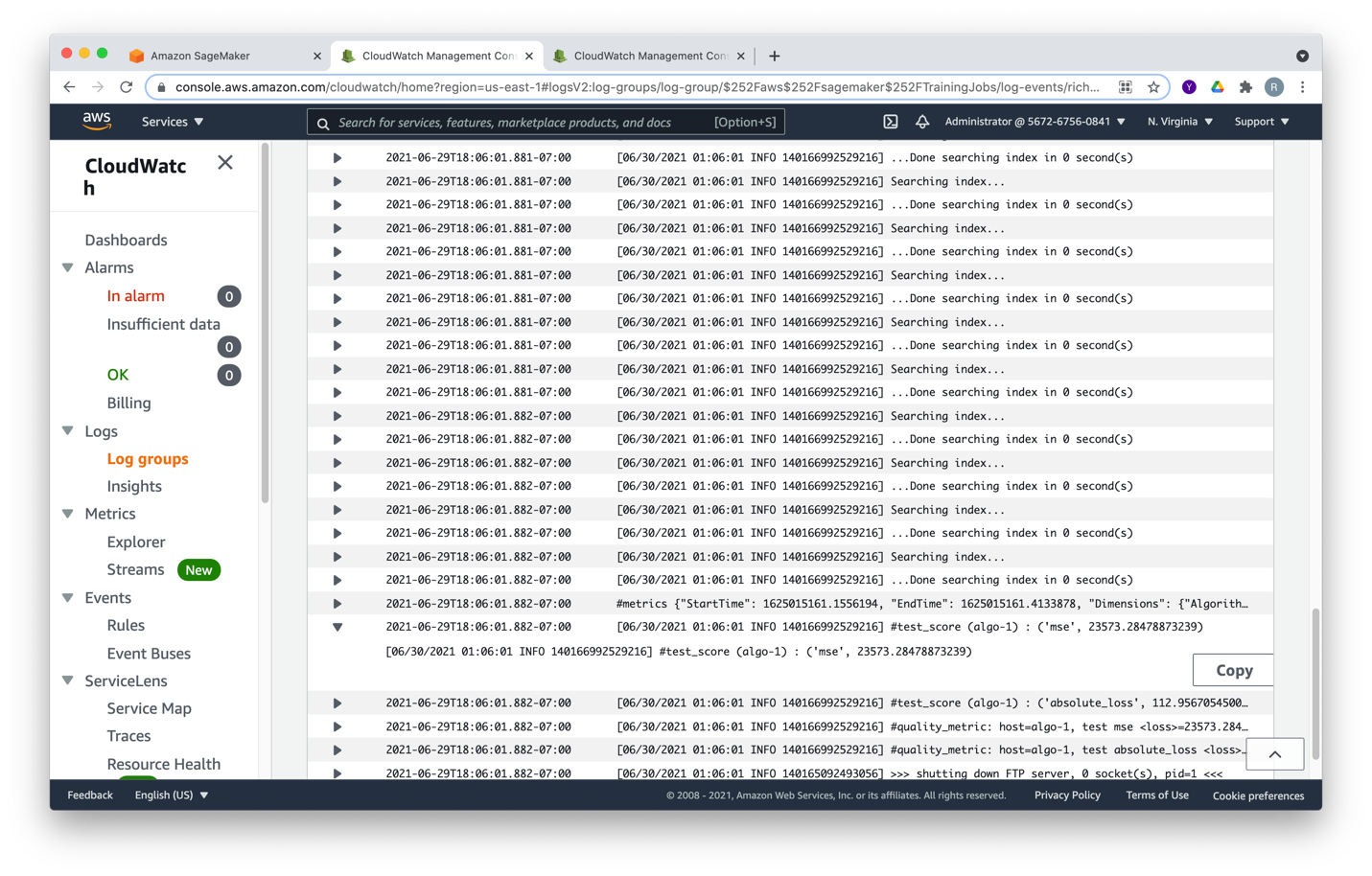


The result of the test is an MSE of 8588 which is an RMSE of 92.67.

You can see this in the output of the log:



Baseline run for one-hot-encoding is done with same hyperparameters. The results are shown in the output log below: MSE = 23573

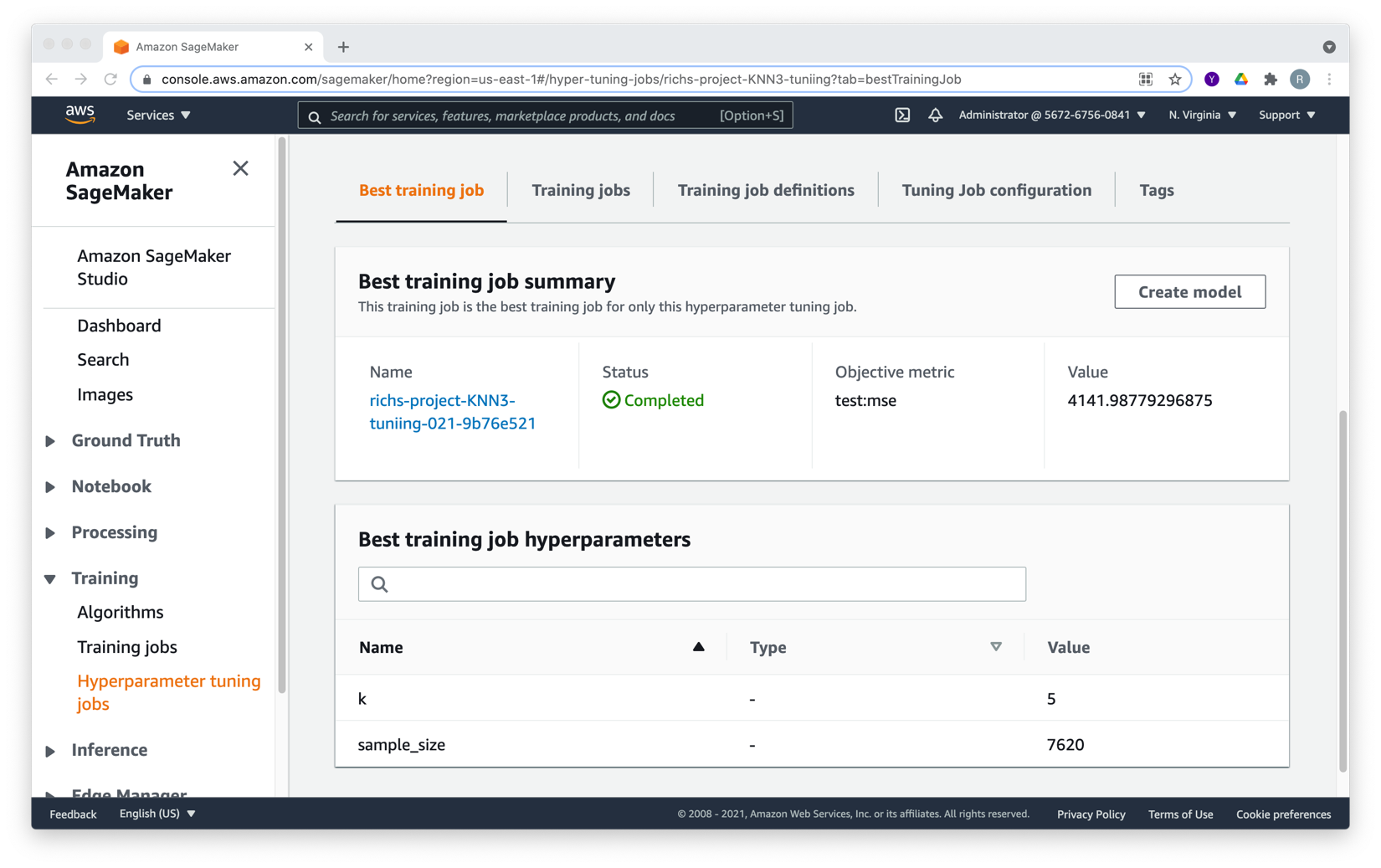


As mentioned in the slides, KNN performed better without one-hot encoding.

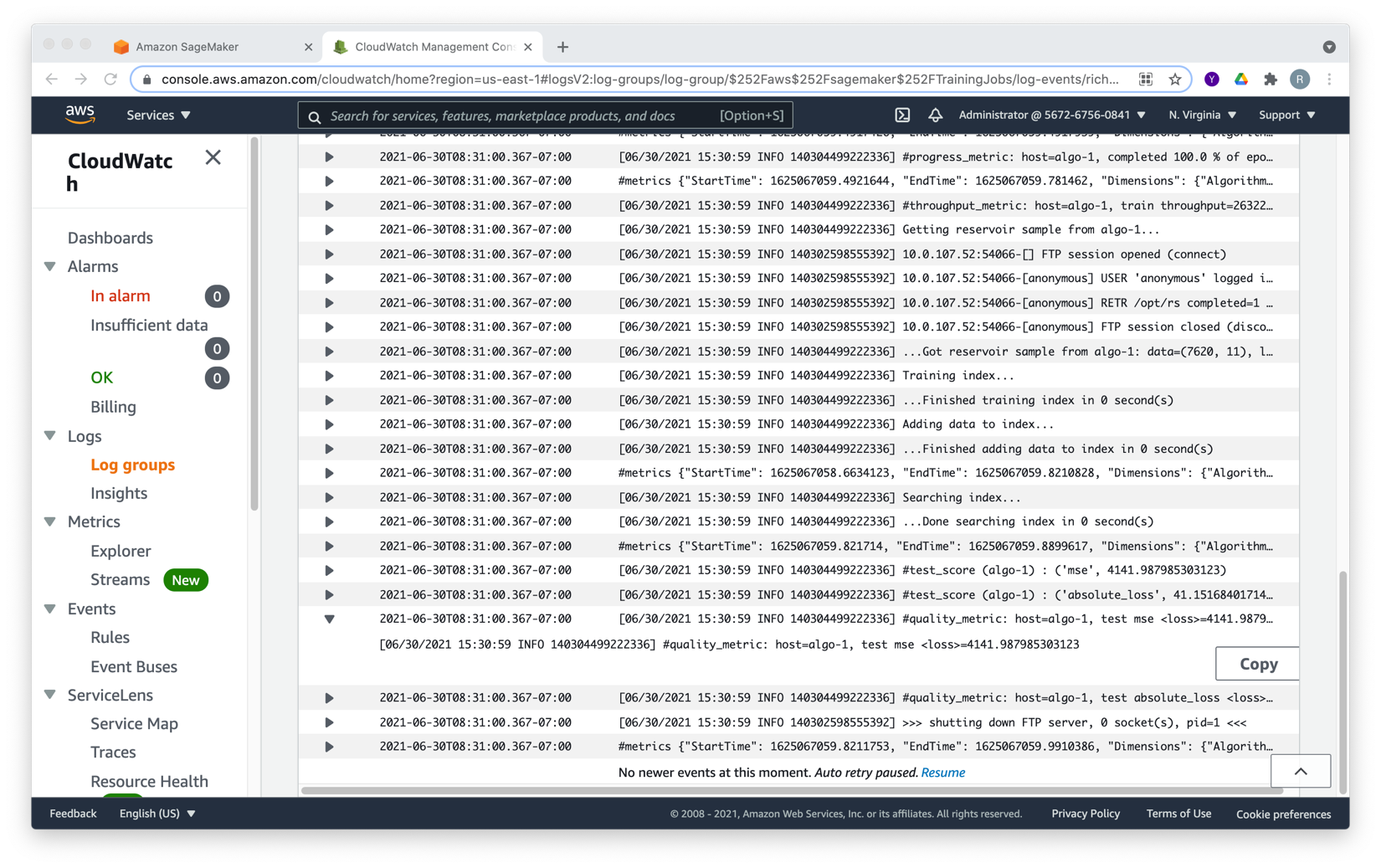
Hyperparameter Tuning KNN

KNN has two hyperparameters to tune K and the sample size.3 hyperparameter tuning jobs were done:

1. A random job to get an understanding of the space with K: 2-10, sample size 1000-2000 over 50 steps with 4 in parallel (always used). The best result was K=6, with sample size = 1905. Looking at the top results, K ranged up to 9 and the sample size was at the high end.
2. Bayesian tuning was used with K: 5-15, sample size: 2000-7600 to ensure K would not grow with the sample size. 12 steps were run. The best result was K=5 and sample size 7588. Looking at top runs, K could be narrowed and can run on whole data set (7620)
3. Bayesian tuning with K = 4-7 and sample size 5000-7620 to be conservative. 36 steps were taken. The best result was with K=5 and sample set = 7620 as shown below:



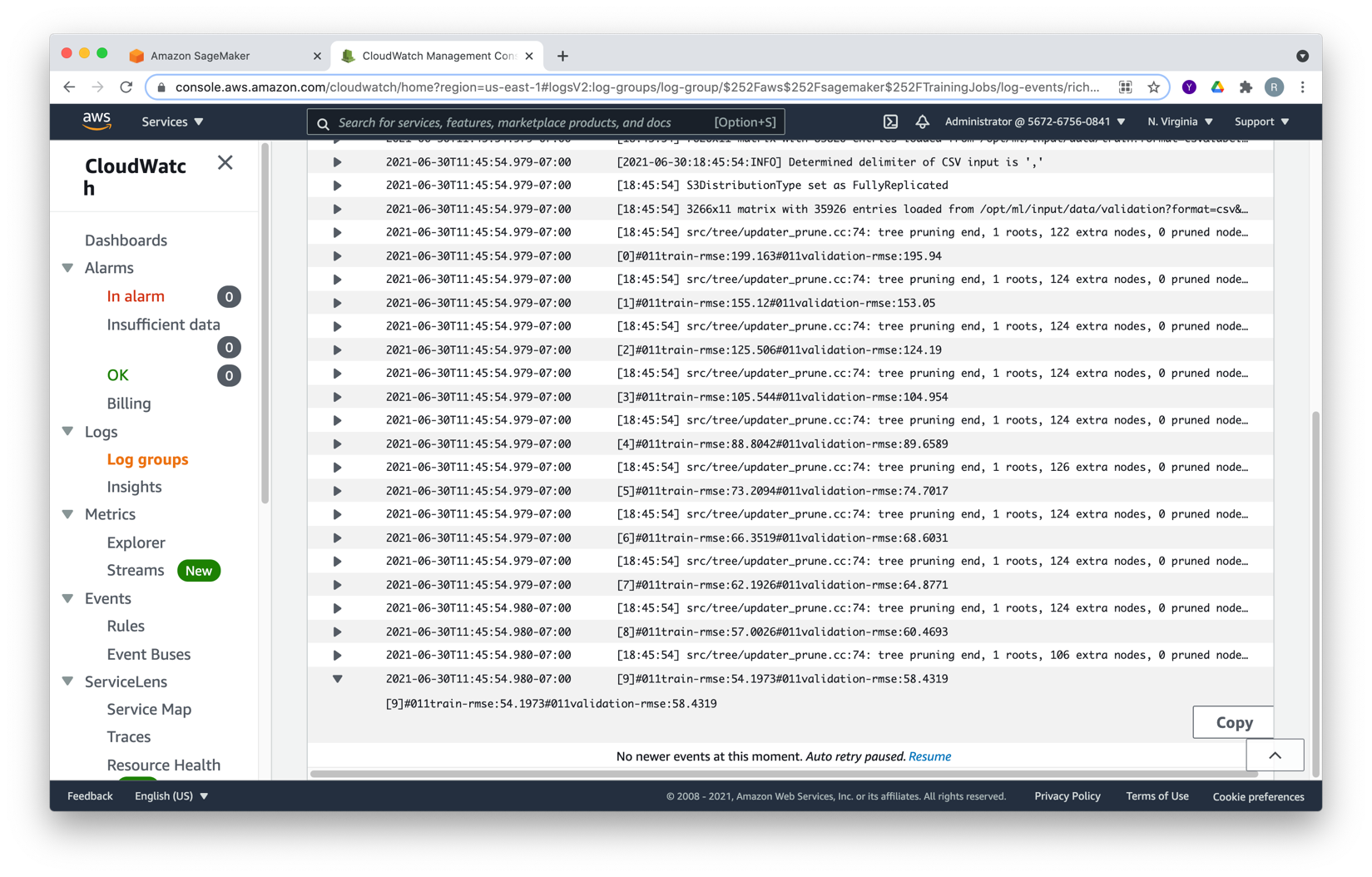
The result with MSE = 4141 is shown below: Tuning was stopped since the top results all had K=5 and sample size at 7620 or near there.



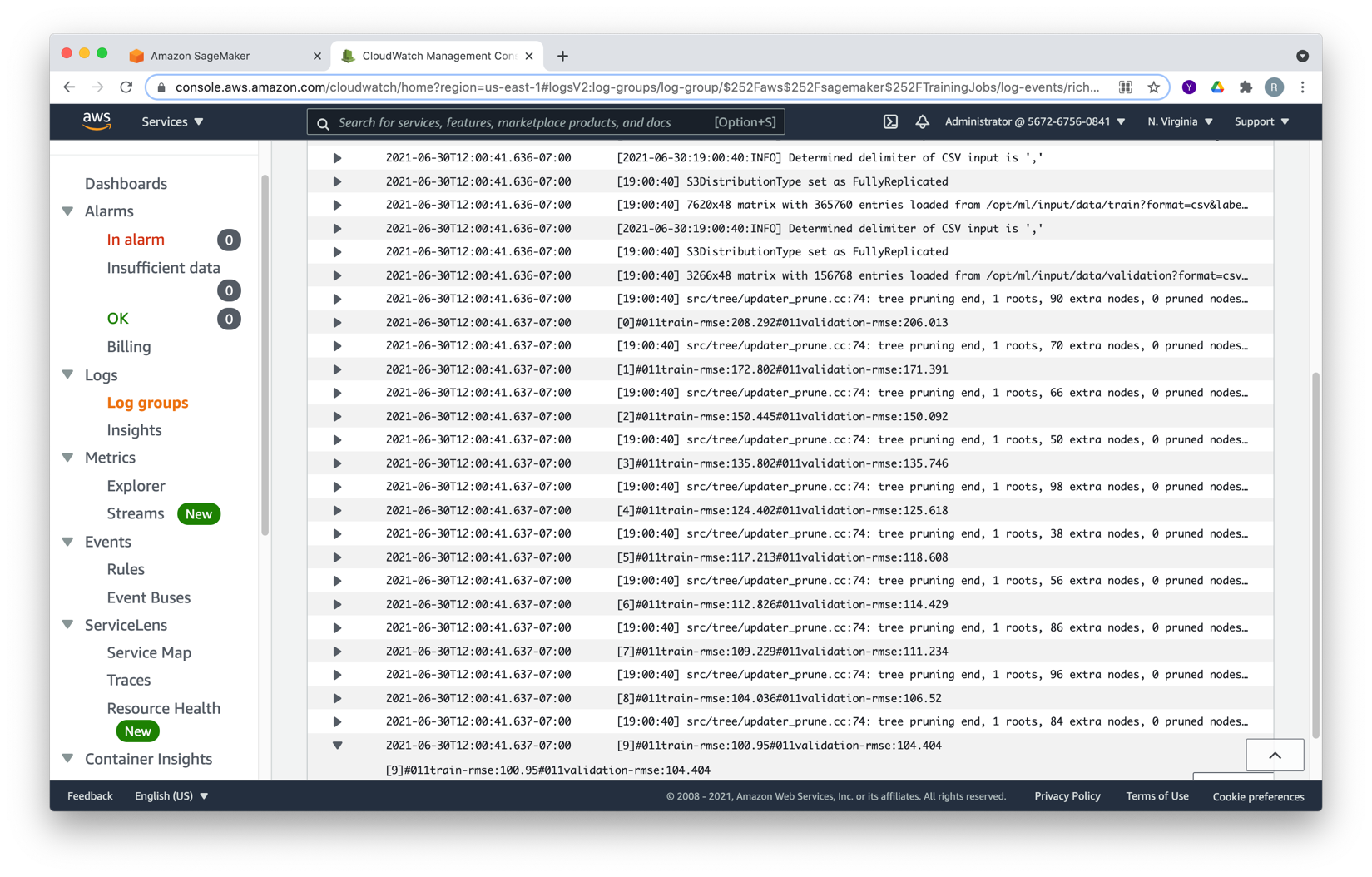
XG Boost

Similar to KNN, 2 baselines were tested with and without one-hot encoding. The baseline used default values with the number of rounds = 10 and max depth = 6.

Without one-hot encoding an RMSE = 58.43 was achieved:



Using the same hyperparameters and one-hot encoding the result was RMSE = 104.40.



As with KNN, XG Boost performs better without one-hot encoding.

XG Boost Hyperparameter Tuning

XG Boost has several tunable parameters. After doing some research online, I decided the number of rounds, max depth and minimum child weight would be the ones to focus on as a start.

3 tuning runs were performed, all using Bayesian optimization.

1. 12 runs with number of rounds: 7-25, max depth 5 -15 and min child weight = 3-10.

The best result was RMSE = 43.02 with number of rounds = 23, max depth = 9 and minimum child weight = 8.19. So, number of rounds should be increased, slight increase in max depth and min child weight can stay the same.

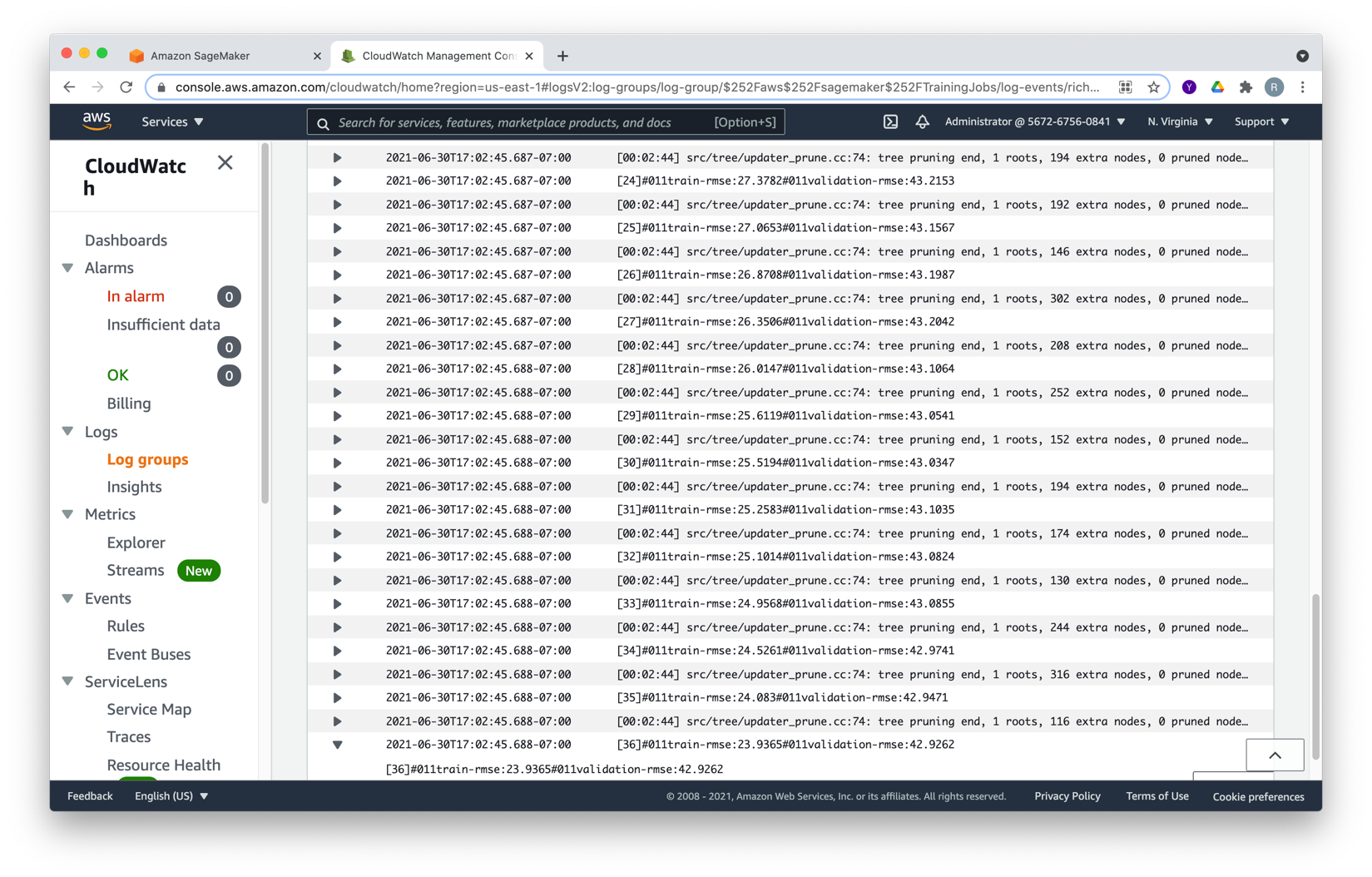
1. 28 runs with number of rounds: 12-30, max depth 7 -17 and min child weight = 3-10.

The best result was RMSE = 43.05 with number of rounds = 30, max depth = 8 and minimum child weight = 4.07. Though the best score was slightly above the previous tuning run, more than the top 6 results were better than the 2nd best in the first tuning round, I continued training. Again, the number of rounds should be increased, max depth and min child weight are stable.

1. 56 runs with number of rounds: 30-50, max depth 7-9 and min child weight = 3-5.

The best result was RMSE = 42.93 with number of rounds = 37, max depth = 8 and minimum child weight = 4.93. A modest improvement was made with only 7 additional rounds so tuning was stopped though this was the 54th run so additional improvement may be possible. Other hyperparameters could be tested as well.

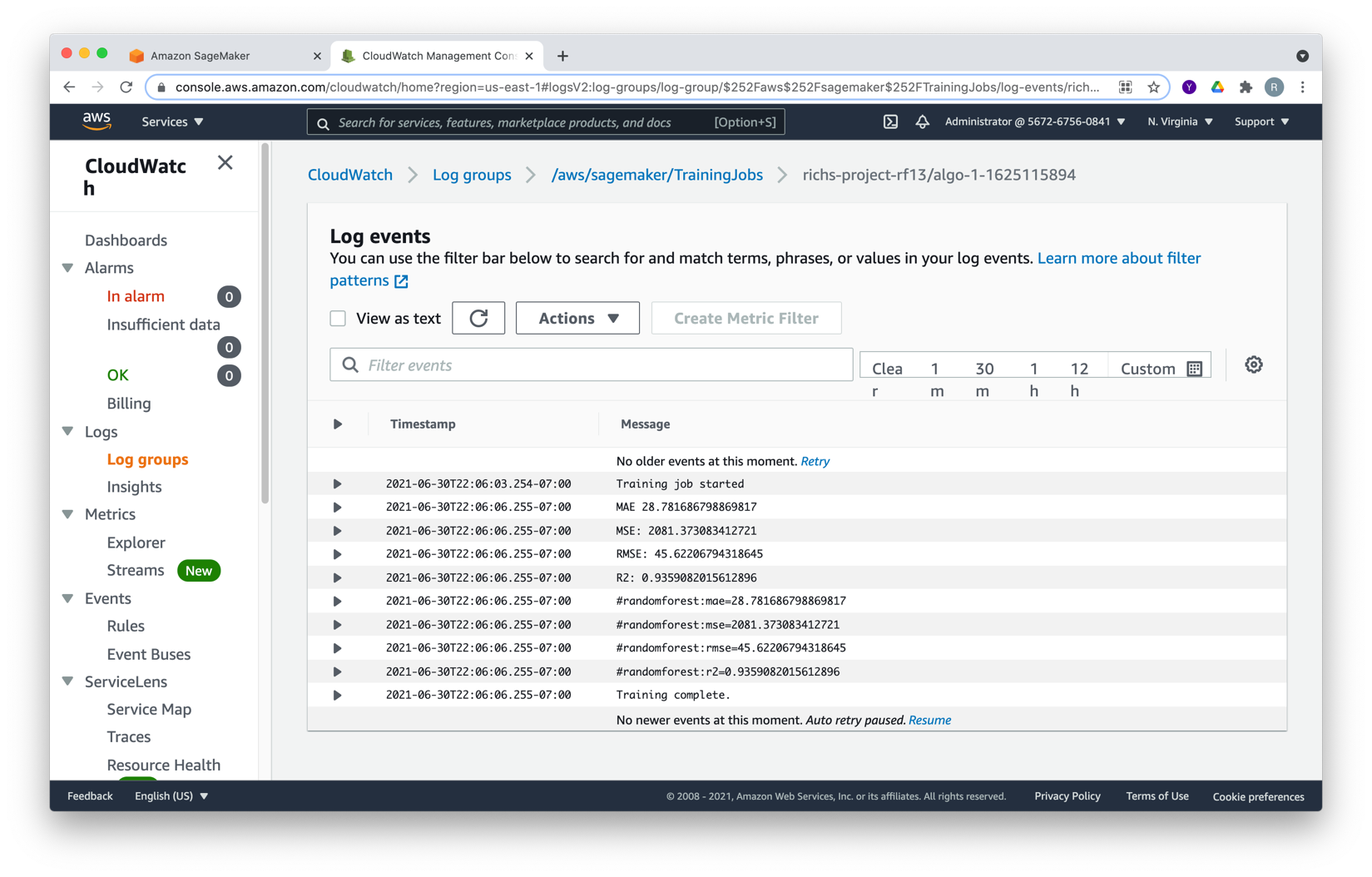
Below is the output log of the best run:



Random Forest (custom)

Since the Random Forest was tested with the Colab notebook with and without one-hot encoding, testing was only done without one-hot encoding as the performance was better. A baseline was done setting the number of estimators to 200 and max depth to 20. The results were:

MAE: 28.78, MSE: 2081, RMSE: 45.62 and R2: 0.94. Below is the output log.



Hyperparameter tuning was attempted however, though the initial jobs completed successfully and reported results to CloudWatch the Hyperparameter Tuning failed with the below message:

**Failure reason**

No objective metrics found after running 5 training jobs. Please ensure that the custom algorithm is emitting the objective metric as defined by the regular expression provided.

I have not been able to figure this out yet and have seen it mentioned in an unanswered question in StackOverFlow.

The first 4-5 jobs did finish for the 2 attempts I made and yielded a slightly better result of RMSE: 45.46 as mentioned in the slides.