Columbia University in the City of New York

AI and OR at Scale on the Cloud

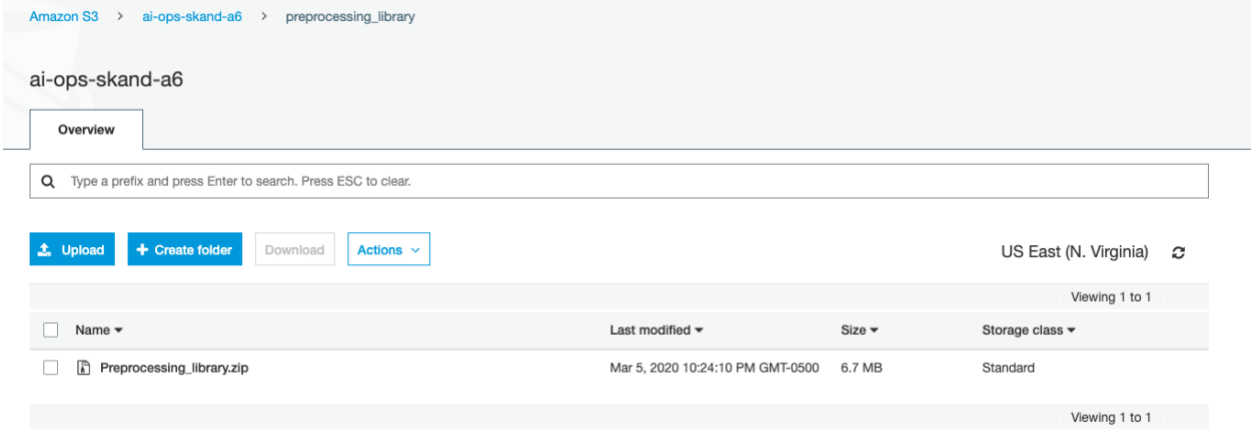
Assignment 6

Group 2: Harsh Mehta (hsm2148) Anunay Sanganal (avs2160) Skand Upmanyu (su2236)

Step 1: Creating a preprocessing library which has the following features:

- clean text
  - remove stopwords
  - removes twitter handles
  - remove URLs
- tokenize text
- create padding
- converts tokens to indices in the dictionary

This preprocessing library is then uploaded to S3:



Step 2: Shuffling the data splitting into 3 parts: Train, Dev and Eval.

Train test split of data

In [1]:
```
import pandas as pd
from sklearn.utils import shuffle
```

In [2]:
```
## Reading data
df = pd.read_csv("training.full.csv")
```

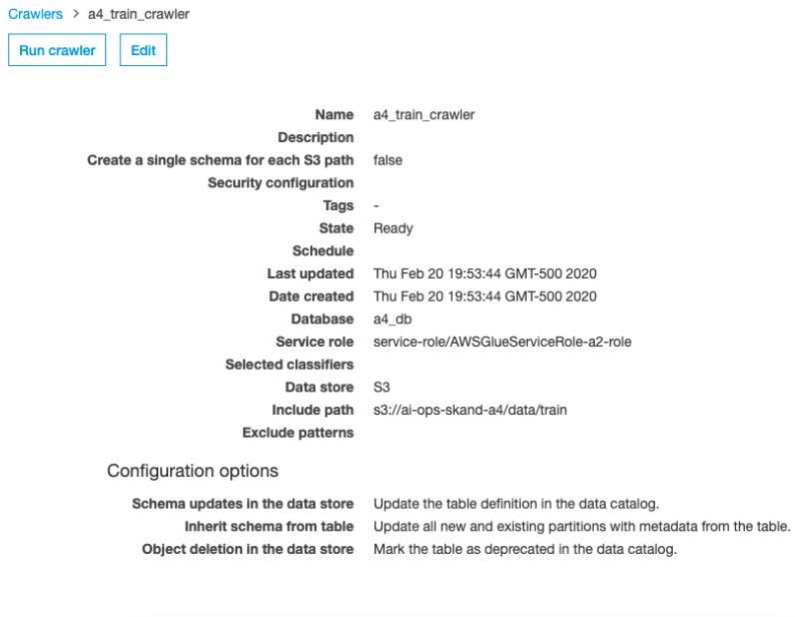
In [3]:
```
## Splitting data
data = shuffle(df)
train = data.iloc[0:int(0.85*len(data)),]
dev = data.iloc[int(0.85*len(data)):int(0.95*len(data)),]
eval = data.iloc[int(0.95*len(data)):,]
```

In [4]:
```
## Writing the data
train.to_csv("training.csv",index=False)
dev.to_csv("dev.csv",index=False)
eval.to_csv("eval.csv",index=False)
```

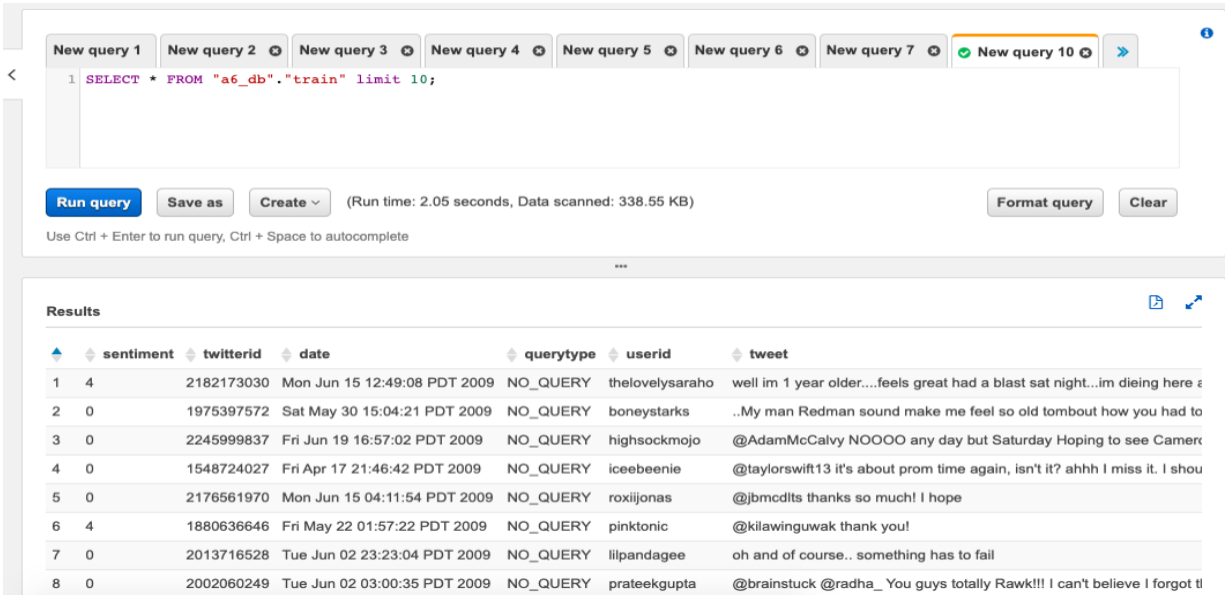
Step 3: Creating a crawler job to create tables for the three datasets

TRAIN:

Crawler for Train dataset



Athena output for train dataset



	sentiment	twitterid	date	querytype	userid	tweet
1	4	2182173030	Mon Jun 15 12:49:08 PDT 2009	NO_QUERY	thelovelysaraho	well im 1 year older....feels great had a blast sat night...im dieing here a
2	0	1975397572	Sat May 30 15:04:21 PDT 2009	NO_QUERY	boneystarks	..My man Redman sound make me feel so old tombout how you had to
3	0	2245999837	Fri Jun 19 16:57:02 PDT 2009	NO_QUERY	highsockmojo	@AdamMcCalvy NOOOO any day but Saturday Hoping to see Camero
4	0	1548724027	Fri Apr 17 21:46:42 PDT 2009	NO_QUERY	iceebeenie	@taylorswift13 it's about prom time again, isn't it? ahhh I miss it. I shou
5	0	2176561970	Mon Jun 15 04:11:54 PDT 2009	NO_QUERY	roxiijonas	@jbmcdlts thanks so much! I hope
6	4	1880636646	Fri May 22 01:57:22 PDT 2009	NO_QUERY	pinktonic	@kilawinguwak thank you!
7	0	2013716528	Tue Jun 02 23:23:04 PDT 2009	NO_QUERY	lilpandagee	oh and of course.. something has to fail
8	0	2002060249	Tue Jun 02 03:00:35 PDT 2009	NO_QUERY	prateekgupta	@brainstuck @radha_ You guys totally Rawk!!! I can't believe I forgot th

EVAL

Crawler for Eval dataset

Crawlers > a6-eval

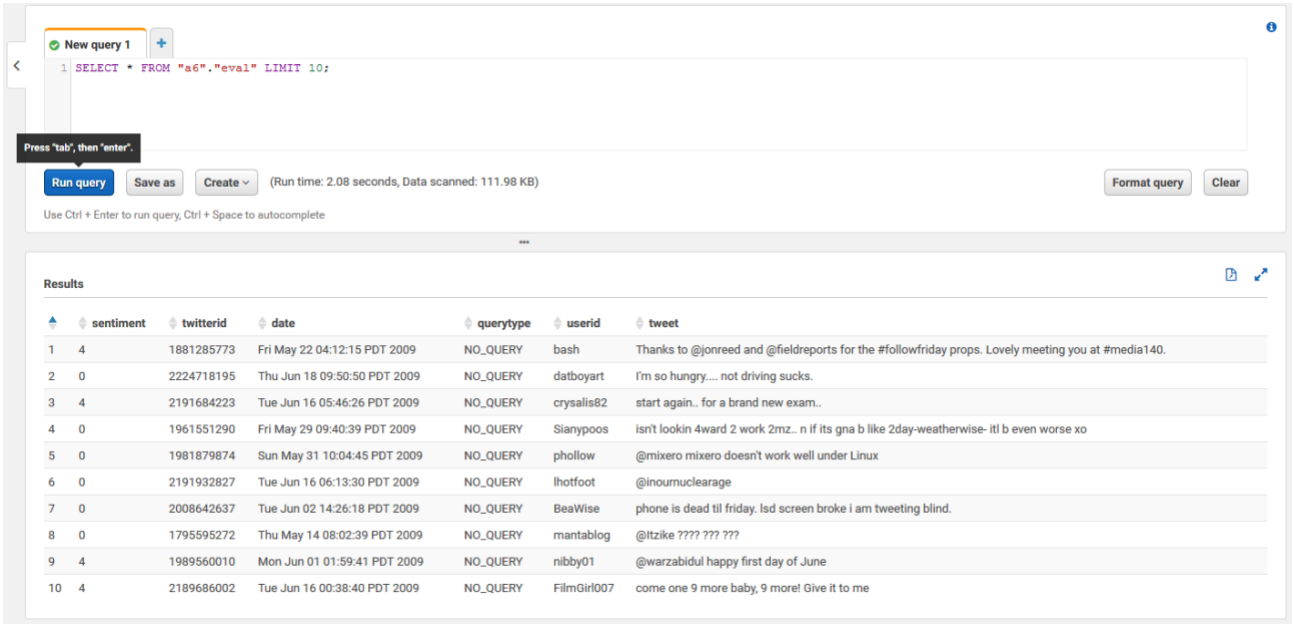
Run crawler | Edit

Name	a6-eval
Description	
Create a single schema for each S3 path	false
Security configuration	
Tags	-
State	Ready
Schedule	
Last updated	Thu Mar 05 21:48:11 GMT-500 2020
Date created	Thu Mar 05 21:48:11 GMT-500 2020
Database	a6
Service role	Assignment_2_twitter
Selected classifiers	
Data store	S3
Include path	s3://aiops-a6/raw_data/eval
Exclude patterns	

Configuration options

Schema updates in the data store	Update the table definition in the data catalog.
Inherit schema from table	Update all new and existing partitions with metadata from the table.
Object deletion in the data store	Mark the table as deprecated in the data catalog.

Athena output for Eval dataset



	sentiment	twitterid	date	querytype	userid	tweet
1	4	1881285773	Fri May 22 04:12:15 PDT 2009	NO_QUERY	bash	Thanks to @jonreed and @fieldreports for the #followfriday props. Lovely meeting you at #media140.
2	0	2224718195	Thu Jun 18 09:50:50 PDT 2009	NO_QUERY	datboyart	I'm so hungry.... not driving sucks.
3	4	2191684223	Tue Jun 16 05:46:26 PDT 2009	NO_QUERY	crysalis82	start again.. for a brand new exam..
4	0	1961551290	Fri May 29 09:40:39 PDT 2009	NO_QUERY	Sianypoos	isn't lookin 4ward 2 work 2mz.. n if its gna b like 2day-weatherwise- itl b even worse xo
5	0	1981879874	Sun May 31 10:04:45 PDT 2009	NO_QUERY	phollow	@mixero mixero doesn't work well under Linux
6	0	2191932827	Tue Jun 16 06:13:30 PDT 2009	NO_QUERY	lhotfoot	@inournuclearage
7	0	2008642637	Tue Jun 02 14:26:18 PDT 2009	NO_QUERY	BeaWise	phone is dead til friday. lsd screen broke i am tweeting blind.
8	0	1795595272	Thu May 14 08:02:39 PDT 2009	NO_QUERY	mantablog	@ltzike ???? ??? ???
9	4	1989560010	Mon Jun 01 01:59:41 PDT 2009	NO_QUERY	nibby01	@warzabidul happy first day of June
10	4	2189686002	Tue Jun 16 00:38:40 PDT 2009	NO_QUERY	FilmGirl007	come one 9 more baby, 9 more! Give it to me

DEV

Crawler for Dev dataset

Crawlers > a6-dev

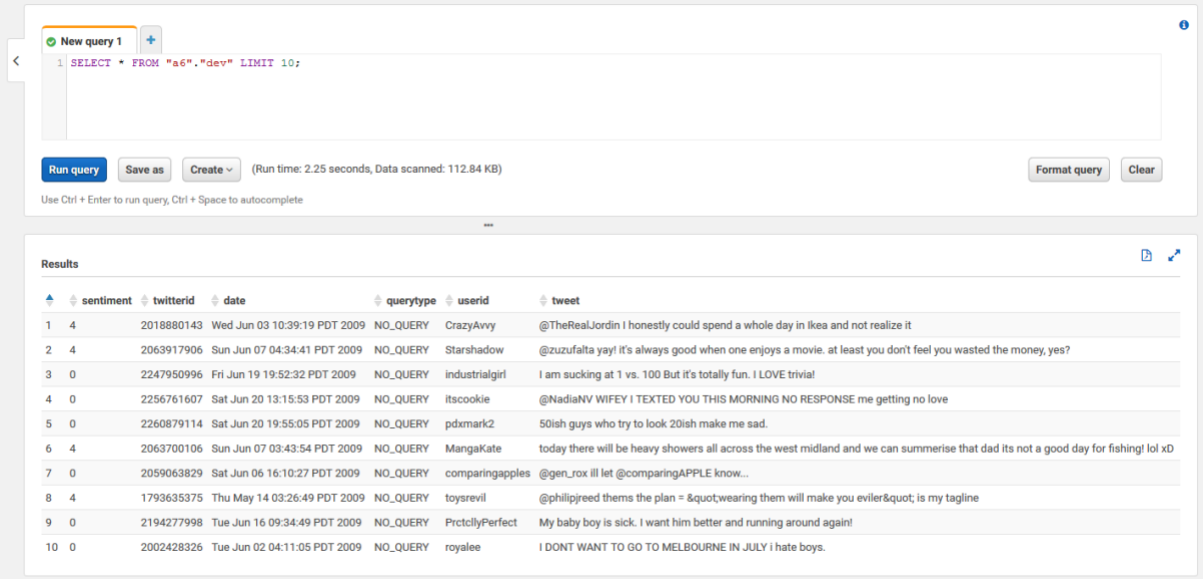
Run crawler | Edit

Name	a6-dev
Description	
Create a single schema for each S3 path	false
Security configuration	
Tags	-
State	Ready
Schedule	
Last updated	Thu Mar 05 21:47:11 GMT-500 2020
Date created	Thu Mar 05 21:47:11 GMT-500 2020
Database	a6
Service role	Assignment_2_twitter
Selected classifiers	
Data store	S3
Include path	s3://aiops-a6/raw_data/dev
Exclude patterns	

Configuration options

Schema updates in the data store	Update the table definition in the data catalog.
Inherit schema from table	Update all new and existing partitions with metadata from the table.
Object deletion in the data store	Mark the table as deprecated in the data catalog.

Athena output for Dev dataset

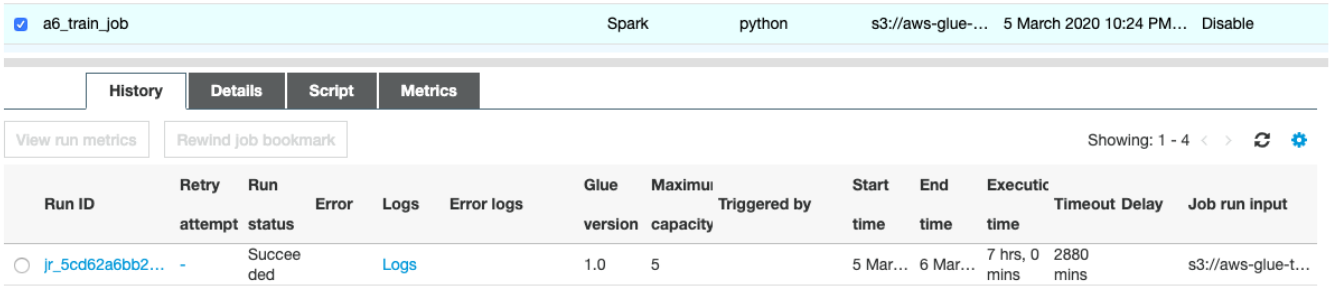


	sentiment	twitterid	date	querytype	userid	tweet
1	4	2018880143	Wed Jun 03 10:39:19 PDT 2009	NO_QUERY	CrazyAvvy	@TheRealJordin I honestly could spend a whole day in Ikea and not realize it
2	4	2063917906	Sun Jun 07 04:34:41 PDT 2009	NO_QUERY	Starshadow	@zuzufalta yay! it's always good when one enjoys a movie. at least you don't feel you wasted the money, yes?
3	0	2247950996	Fri Jun 19 19:52:32 PDT 2009	NO_QUERY	industrialgirl	I am sucking at 1 vs. 100 But it's totally fun. I LOVE trivia!
4	0	2256761607	Sat Jun 20 13:15:53 PDT 2009	NO_QUERY	itscookie	@NadiaNV WIFEY I TEXTED YOU THIS MORNING NO RESPONSE me getting no love
5	0	2260879114	Sat Jun 20 19:55:05 PDT 2009	NO_QUERY	pdxmark2	50ish guys who try to look 20ish make me sad.
6	4	2063700106	Sun Jun 07 03:43:54 PDT 2009	NO_QUERY	MangaKate	today there will be heavy showers all across the west midland and we can summerise that dad its not a good day for fishing! lol xD
7	0	2059063829	Sat Jun 06 16:10:27 PDT 2009	NO_QUERY	comparingapples	@gen_rox ill let @comparingAPPLE know...
8	4	1793635375	Thu May 14 03:26:49 PDT 2009	NO_QUERY	toysrevil	@philipjreed thems the plan = "wearing them will make you eviler" is my tagline
9	0	2194277998	Tue Jun 16 09:34:49 PDT 2009	NO_QUERY	PrctcllyPerfect	My baby boy is sick. I want him better and running around again!
10	0	2002428326	Tue Jun 02 04:11:05 PDT 2009	NO_QUERY	royalee	I DONT WANT TO GO TO MELBOURNE IN JULY i hate boys.

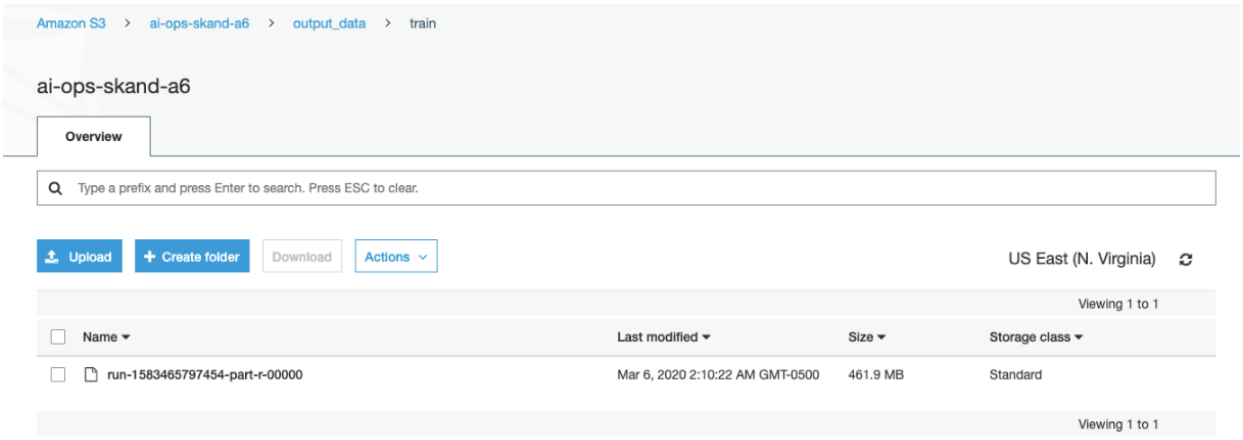
Step 4: Running a Glue job to create features using the pre-processing library

TRAIN

Successful completion of the Glue job

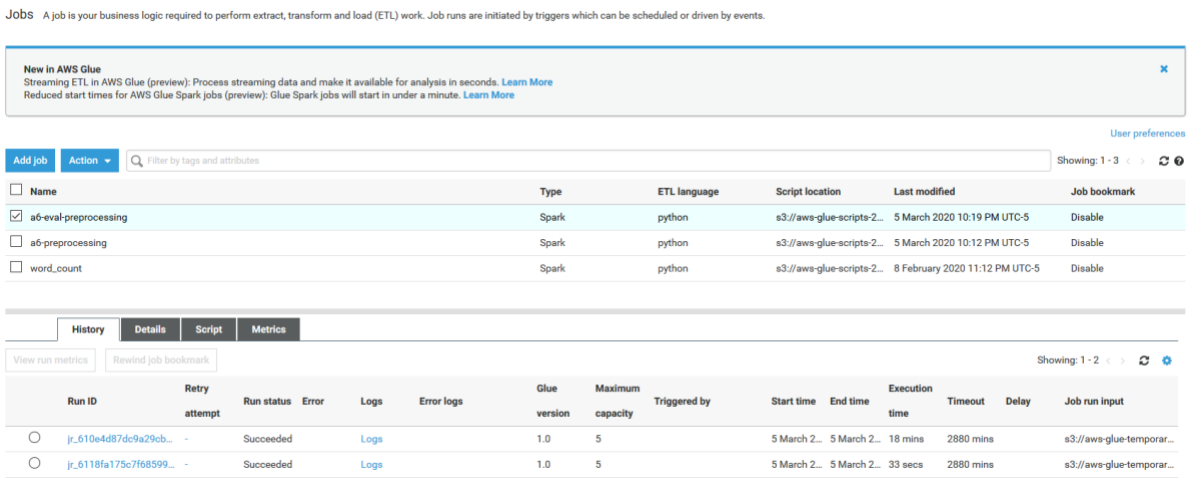


The JSON file generated in S3

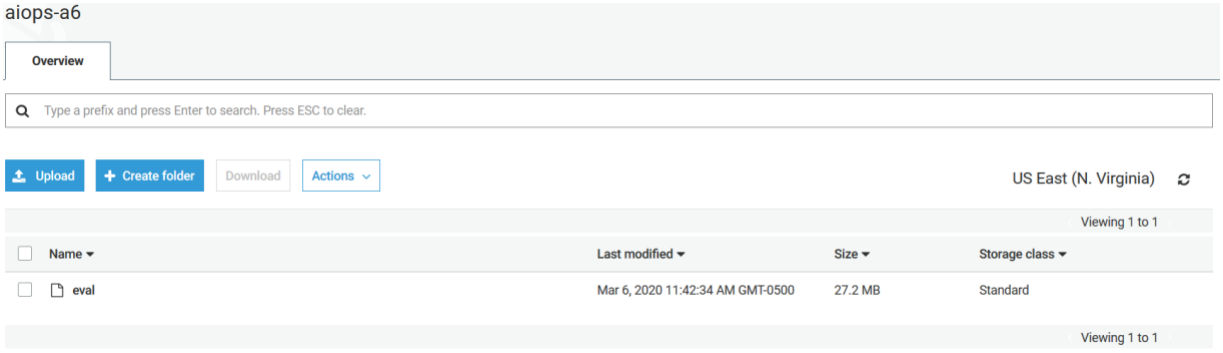


EVAL

Successful completion of the Glue job

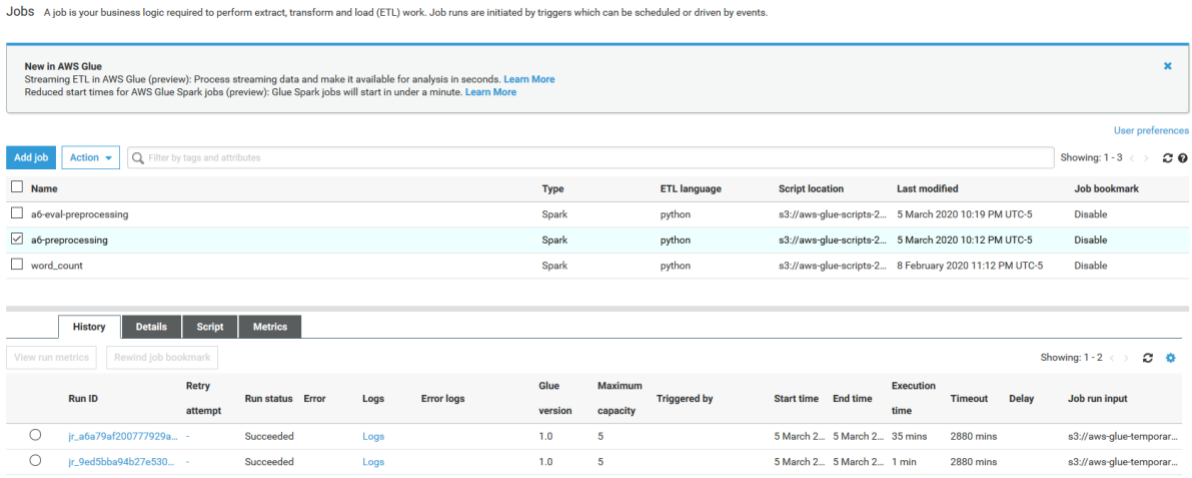


The JSON file generated in S3

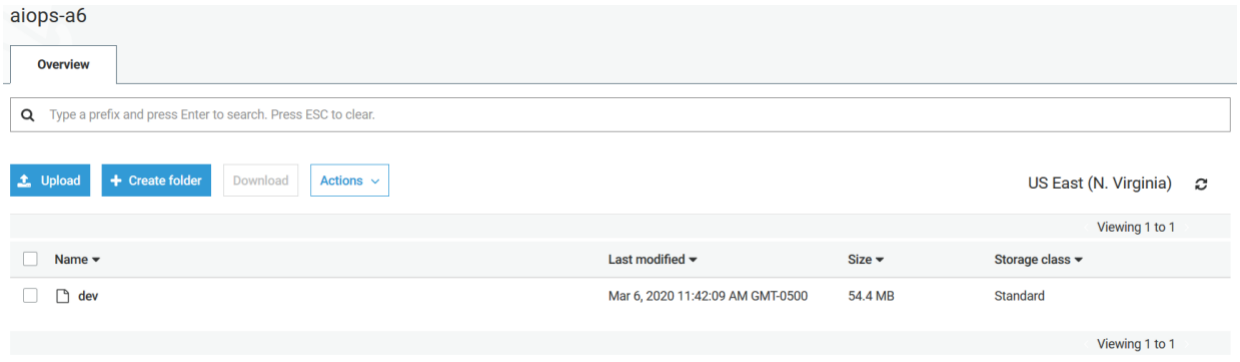


DEV

Successful completion of the Glue job



The JSON file generated in S3



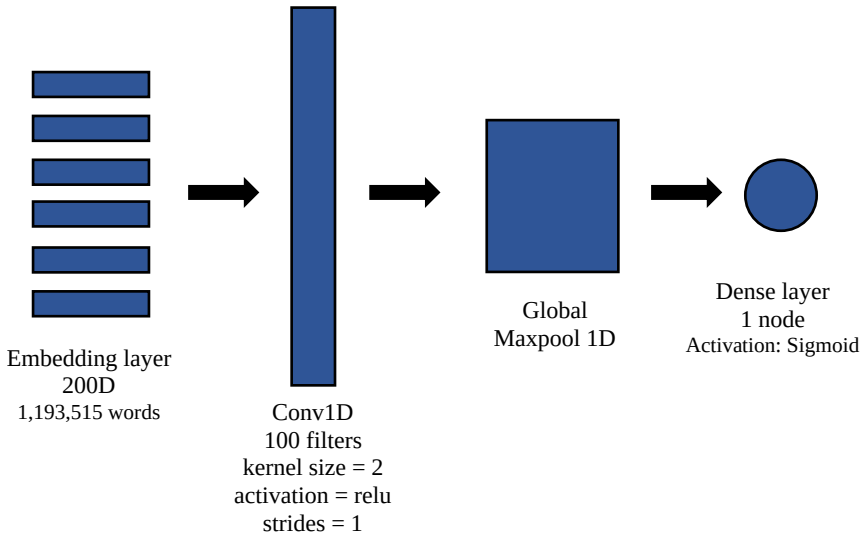
Step 5: Model training :

We tried various models which belong to 5 distinct categories:

1. CNN: as given in assignment 4 (we treated this model as the benchmark)
2. Dense Neural Network
3. Deep CNN
4. GRU (Gated recurrent unit)
5. LSTM (Long Short Term Memory)

Let us look into these models in detail:

1. CNN: as given in assignment 4 (we treated this model as the benchmark)



Learning rate = 0.0005

Loss = Binary cross-entropy

Optimizer = Adam

Epochs = 3

Batch size = 1000

Training accuracy = 0.8186

Validation accuracy = 0.7941

Test accuracy = 0.7941

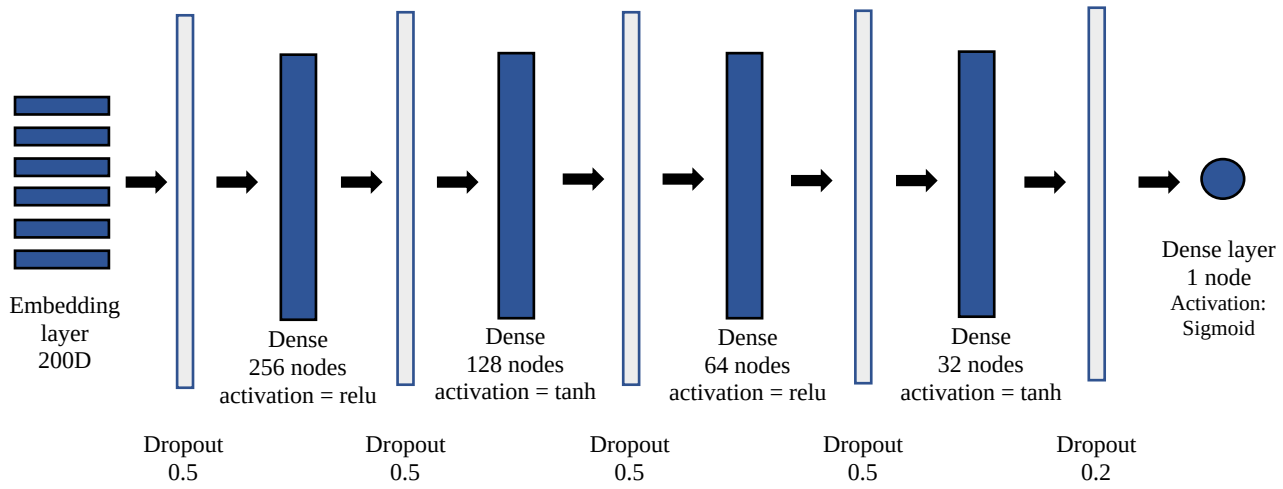
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 100, 200)          238703000
_________________________________________________________________
conv1d (Conv1D)              (None, 99, 100)           40100
_________________________________________________________________
global_max_pooling1d (Global (None, 100)               0
_________________________________________________________________
dense (Dense)                (None, 100)               10100
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 101
=================================================================
Total params: 238,753,301
Trainable params: 238,753,301
Non-trainable params: 0
_________________________________________________________________
```

```
Train on 1360 samples, validate on 1000 samples
Epoch 1/3
1360/1360 [==============================] - 346s 255ms/step - loss: 0.4635 - acc: 0.7766 - val_loss: 0.4426 - val_ac
c: 0.7911
Epoch 2/3
1360/1360 [==============================] - 324s 238ms/step - loss: 0.4221 - acc: 0.8033 - val_loss: 0.4378 - val_ac
c: 0.7944
Epoch 3/3
1360/1360 [==============================] - 323s 238ms/step - loss: 0.3966 - acc: 0.8186 - val_loss: 0.4399 - val_ac
c: 0.7941
Test loss:0.4395864550024271
Test accuracy:0.7940875291824341
```

Hyperparameters tuned:

- Learning rate: selected a learning rate which was not too high and too low. Tried different values between 0.00001 to 0.005
- Batch size: a batch size of 1000 gave us better computational performance and fast convergence
- Epochs: We trained for more epochs and observed when the validation accuracy started decreasing. The no. of epochs with the highest validation accuracy was then selected
- We did not change the architecture in this case as this was our benchmark model

2. Dense Neural Network



Learning rate = 0.0005

Loss = Binary cross-entropy

Optimizer = Adam

Epochs = 3

Batch size = 1000

Training accuracy = 0.7982

Validation accuracy = 0.7885

Test accuracy = 0.7865

Hyperparameters tuned:

- #Layers, nodes, activation, dropout %: tried different architectures and selected the one with the highest validation accuracy
- Learning rate: selected a learning rate which was not too high and too low. Tried different values between 0.00001 to 0.005
- Batch size: a batch size of 1000 gave us better computational performance and fast convergence
- Optimizer: Adam optimizer helped us achieve faster convergence
- Epochs: We trained for more epochs and observed when the validation accuracy started decreasing. The no. of epochs with the highest validation accuracy was then selected

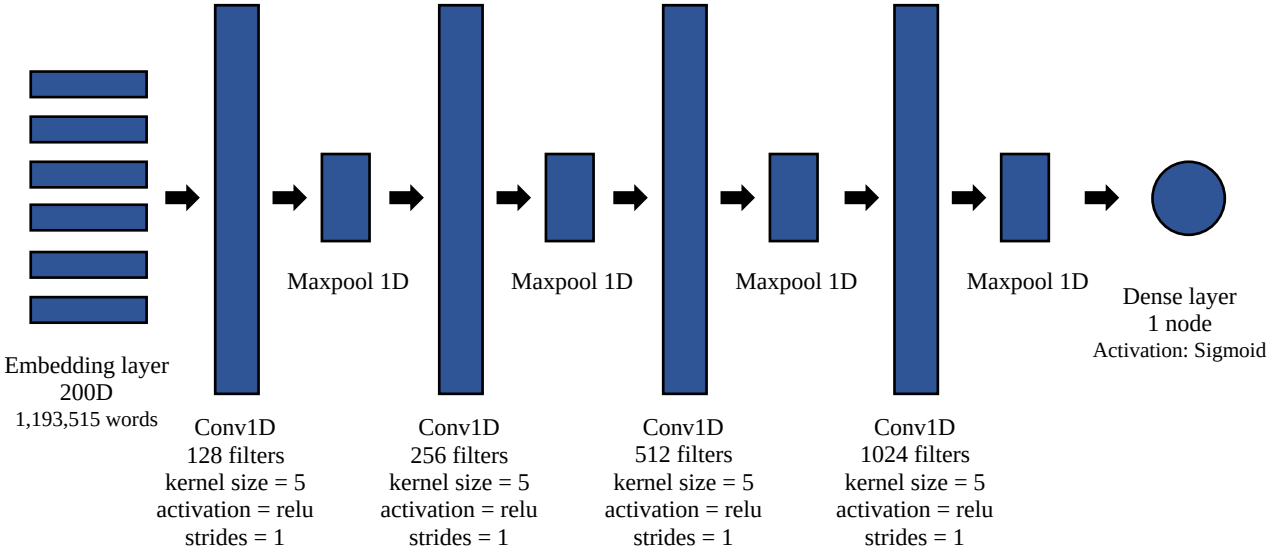
Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 100, 200)	238703000
flatten (Flatten)	(None, 20000)	0
dense (Dense)	(None, 256)	5120256
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
dropout_2 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2080
dropout_3 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 1)	33

Total params: 243,866,521
Trainable params: 243,866,521
Non-trainable params: 0

```
Train on 1360 samples, validate on 1000 samples
Epoch 1/3
1360/1360 [==============================] - 316s 232ms/step - loss: 0.4979 - acc: 0.7587 - val_loss: 0.4585 - val_ac
c: 0.7836
Epoch 2/3
1360/1360 [==============================] - 299s 220ms/step - loss: 0.4545 - acc: 0.7880 - val_loss: 0.4545 - val_ac
c: 0.7865
Epoch 3/3
1360/1360 [==============================] - 299s 220ms/step - loss: 0.4369 - acc: 0.7982 - val_loss: 0.4518 - val_ac
c: 0.7885
Test loss:0.45280549079179766
Test accuracy:0.7864999771118164
```

3. Deep CNN



Learning rate = 0.0005

Loss = Binary cross-entropy

Optimizer = Adam

Epochs = 2

Batch size = 1000

Training accuracy = 0.8066

Validation accuracy = 0.7974

Test accuracy = 0.7974

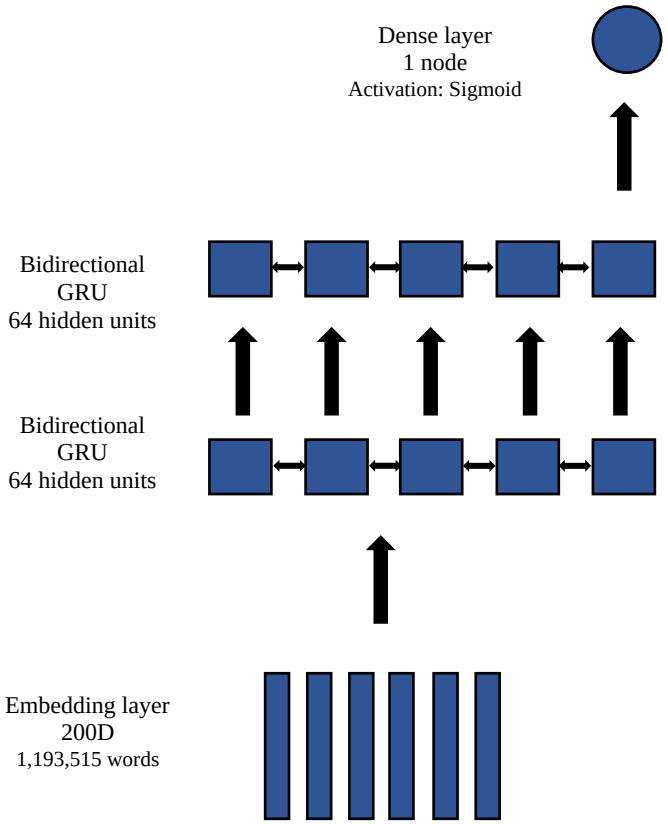
Hyperparameters tuned:

- #Layers, nodes, activation: tried different architectures and selected the one with the highest validation accuracy
- Learning rate: selected a learning rate which was not too high and too low. Tried different values between 0.00001 to 0.005
- Batch size: a batch size of 1000 gave us better computational performance and fast convergence
- Optimizer: Adam optimizer helped us achieve faster convergence
- Epochs: We trained for more epochs and observed when the validation accuracy started decreasing. The no. of epochs with the highest validation accuracy was then selected

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 100, 200)          238703000
_________________________________________________________________
conv1d (Conv1D)              (None, 96, 128)           128128
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 48, 128)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 44, 256)           164096
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 22, 256)           0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 18, 512)           655872
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 9, 512)            0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 5, 1024)           2622464
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 2, 1024)           0
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 1024)              2098176
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 1025
=================================================================
Total params: 244,372,761
Trainable params: 244,372,761
Non-trainable params: 0
_________________________________________________________________
```

```
Train on 1360 samples, validate on 1000 samples
Epoch 1/2
1360/1360 [==============================] - 580s 427ms/step - loss: 0.4583 - acc: 0.7804 - val_loss: 0.4381 - val_ac
c: 0.7928
Epoch 2/2
1360/1360 [==============================] - 559s 411ms/step - loss: 0.4169 - acc: 0.8066 - val_loss: 0.4323 - val_ac
c: 0.7974
Test loss:0.4322528585791588
Test accuracy:0.7974374890327454
```

4. GRU (Gated recurrent unit)



Learning rate = 0.0005
Loss = Binary cross-entropy
Optimizer = Adam
Epochs = 4
Batch size = 1000
Training accuracy = 0.8009
Validation accuracy = 0.7963
Test accuracy = 0.7947

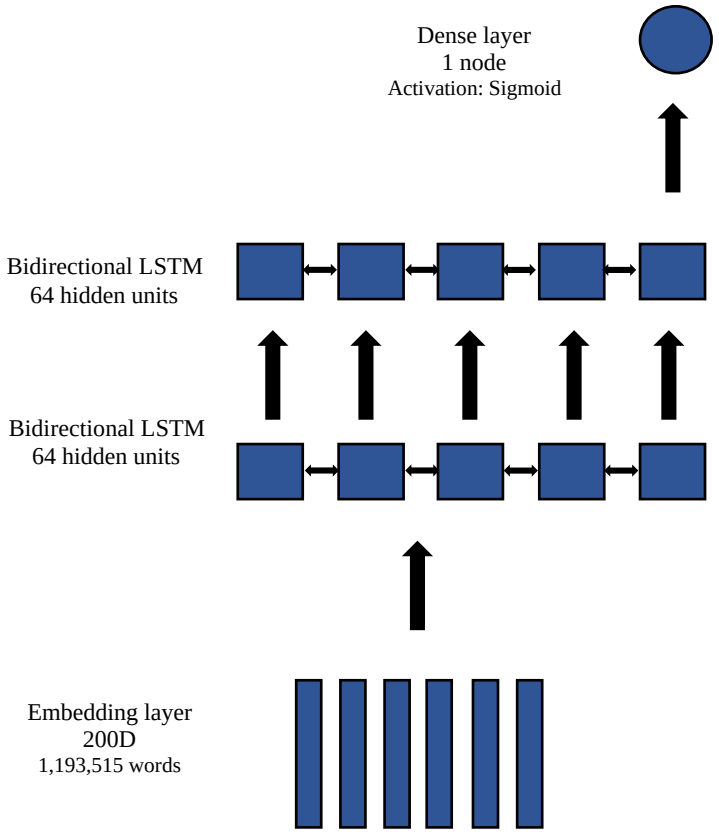
Hyperparameters tuned:

- Bidirectional, single direction, hidden units, dropout %: tried different architectures and selected the one with the highest validation accuracy
- Learning rate: selected a learning rate which was not too high and too low. Tried different values between 0.00001 to 0.005
- Batch size: a batch size of 1000 gave us better computational performance and fast convergence
- Optimizer: Adam optimizer helped us achieve faster convergence
- Epochs: We trained for more epochs and observed when the validation accuracy started decreasing. The no. of epochs with the highest validation accuracy was then selected

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 100, 200)          238703000
_________________________________________________________________
bidirectional (Bidirectional (None, 100, 128)          101760
_________________________________________________________________
bidirectional_1 (Bidirection (None, 128)               74112
_________________________________________________________________
dense (Dense)                (None, 1)                 129
=================================================================
Total params: 238,879,001
Trainable params: 238,879,001
Non-trainable params: 0
_________________________________________________________________
```

```
Train on 1360 samples, validate on 1000 samples
Epoch 1/4
1360/1360 [==============================] - 1381s 1s/step - loss: 0.4950 - acc: 0.7566 - val_loss: 0.4497 - val_acc:
0.7882
Epoch 2/4
1360/1360 [==============================] - 1353s 995ms/step - loss: 0.4514 - acc: 0.7864 - val_loss: 0.4415 - val_a
cc: 0.7929
Epoch 3/4
1360/1360 [==============================] - 1354s 996ms/step - loss: 0.4376 - acc: 0.7949 - val_loss: 0.4389 - val_a
cc: 0.7944
Epoch 4/4
1360/1360 [==============================] - 1352s 994ms/step - loss: 0.4271 - acc: 0.8009 - val_loss: 0.4374 - val_a
cc: 0.7963
Test loss:0.4378286588937044
Test accuracy:0.7947624921798706
```

5. LSTM (Long Short Term Memory)



Learning rate = 0.0005
Loss = Binary cross-entropy
Optimizer = Adam
Epochs = 5
Batch size = 1000
Training accuracy = 0.8084
Validation accuracy = 0.7969
Test accuracy = 0.7961

- Bidirectional, single direction, hidden units, dropout %: tried different architectures and selected the one with the highest validation accuracy
- Learning rate: selected a learning rate which was not too high and too low. Tried different values between 0.00001 to 0.005
- Batch size: a batch size of 1000 gave us better computational performance and fast convergence
- Optimizer: Adam optimizer helped us achieve faster convergence
- Epochs: We trained for more epochs and observed when the validation accuracy started decreasing. The no. of epochs with the highest validation accuracy was then selected

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 100, 200)          238703000
_________________________________________________________________
bidirectional (Bidirectional (None, 100, 128)          135680
_________________________________________________________________
bidirectional_1 (Bidirection (None, 128)               98816
_________________________________________________________________
dense (Dense)                (None, 1)                 129
=================================================================
Total params: 238,937,625
Trainable params: 238,937,625
Non-trainable params: 0
_________________________________________________________________
```

```
Train on 1360 samples, validate on 1000 samples
Epoch 1/5
1360/1360 [==============================] - 1316s 968ms/step - loss: 0.4941 - acc: 0.7577 - val_loss: 0.4514 - val_a
cc: 0.7869
Epoch 2/5
1360/1360 [==============================] - 1264s 929ms/step - loss: 0.4499 - acc: 0.7871 - val_loss: 0.4419 - val_a
cc: 0.7928
Epoch 3/5
1360/1360 [==============================] - 1285s 945ms/step - loss: 0.4345 - acc: 0.7966 - val_loss: 0.4390 - val_a
cc: 0.7941
Epoch 4/5
1360/1360 [==============================] - 1250s 919ms/step - loss: 0.4233 - acc: 0.8028 - val_loss: 0.4366 - val_a
cc: 0.7964
Epoch 5/5
1360/1360 [==============================] - 1291s 949ms/step - loss: 0.4141 - acc: 0.8084 - val_loss: 0.4374 - val_a
cc: 0.7969
Test loss:0.43729878440499304
Test accuracy:0.7961750030517578
```

The best validation accuracy was achieved using Deep CNN

Sagemaker training screenshots:

Model1: CNN Model from Assignment 4 (benchmark)

In [ ]:
```
# Define estimator
estimator = TensorFlow(base_job_name='a6', \
                       entry_point='sentiment_training.py', \
                       source_dir='s3://ai-ops-skand-a6/model_training/model_training1_CNN.tar.gz', \
                       role=role, \
                       framework_version='1.14.0', py_version='py3', \
                       hyperparameters={'num_epoch' : 2,
                                        'config_file': 'training_config.json',
                                        'model_dir': 's3://ai-ops-skand-a6/models/test_model'}, \
                       train_instance_count=1, train_instance_type='ml.c4.8xlarge')

# Fit estimator
estimator.fit({'train' : 's3://ai-ops-skand-a6/model_data/train', \
               'validation' : 's3://ai-ops-skand-a6/model_data/dev', \
               'eval' : 's3://ai-ops-skand-a6/model_data/eval'})
```

```
2020-03-09 03:17:14 Starting - Starting the training job...
2020-03-09 03:17:15 Starting - Launching requested ML instances......
2020-03-09 03:18:18 Starting - Preparing the instances for training......
2020-03-09 03:19:22 Downloading - Downloading input data...
2020-03-09 03:20:01 Training - Training image download completed. Training in progress./usr/local/lib/python3.6/site-
packages/tensorflow/python/framework/dtypes.py:516: FutureWarning: Passing (type, 1) or '1type' as a synonym of type
is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/usr/local/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:517: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) /
'(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/usr/local/lib/python3.6/site-packages/tensorflow/python/framework/dtypes.py:518: FutureWarning: Passing (type, 1) or
'1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) /
'(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
```

Model1: CNN Model from Assignment 4 (benchmark)

In [ ]:
```
# Define estimator
estimator = TensorFlow(base_job_name='a6', \
                       entry_point='sentiment_training.py', \
                       source_dir='s3://ai-ops-skand-a6/model_training/model_training1_CNN.tar.gz', \
                       role=role, \
                       framework_version='1.14.0', py_version='py3', \
                       hyperparameters={'num_epoch' : 2,
                                        'config_file': 'training_config.json',
                                        'model_dir': 's3://ai-ops-skand-a6/models/test_model'}, \
                       train_instance_count=1, train_instance_type='ml.c4.8xlarge')

# Fit estimator
estimator.fit({'train' : 's3://ai-ops-skand-a6/model_data/train', \
               'validation' : 's3://ai-ops-skand-a6/model_data/dev', \
               'eval' : 's3://ai-ops-skand-a6/model_data/eval'})
```

```
010#010#010#010#010#010#0151358/1360 [============================>.] - ETA: 2s - loss: 0.4221 - acc: 0.8035#010#010#
010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#0
10#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#01
0#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#0151359/1360 [==========================
=======>.] - ETA: 1s - loss: 0.4221 - acc: 0.8035#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010
#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#
010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#010#0
10#010#010#010#010#010#0151360/1360 [==============================] - 1459s 1s/step - loss: 0.4221 - acc: 0.8035 - v
al_loss: 0.4375 - val_acc: 0.7944

2020-03-09 04:25:32 Uploading - Uploading generated training modelTest loss:0.4381387535482645
Test accuracy:0.7933499813079834
2020-03-09 04:25:21.596506: W tensorflow/python/util/util.cc:280] Sets are not currently considered sequences, but th
is may change in the future, so consider avoiding using them.
Model successfully saved at: /opt/ml/model/sentiment_model.h5

2020-03-09 04:28:14 Completed - Training job completed
Training seconds: 4132
Billable seconds: 4132
```

Misclassification:

Next, we manually checked which tweets are getting misclassified to improve the model further (if there is scope)

Tweet	Prediction	Actual Sentiment	Comments
@Itzike ???? ??? ???	0.76422673	0	*No sentiment*
Grill in the oven is almost as good as on a grill. But I have no yard to grill in	0.95532244	0	*Mixed emotions*
Reading reviews.. One big flaw, only 8GB of storage on the palm. LAF!! Roll on iPhone release - 32GB on board. Defo not iPhone killer. X	0.2646958	1	*Confusing*
thinking of you..	0.51759243	0	*Subject to interpretation*
I am so bored, someone entertain me?	0.2892712	1	*Neutral*
No more ice	0.72411424	0	*Subject to interpretation*
wow the things that could happen to LeBRON James. SMH. lol.	0.26916614	1	*Confusing SMH and lol together*
m off... studying again	0.046333984	1	*May be mislabeled*
First day of finals and I don't have any	0.47228208	1	*Confusing*
@kirawr OMG..and you didnt tell me this before..Hhmm?!?! I cant see not zombie siching on you tho bwahahaha i love you	0.52002203	0	*Mixed emotions*
@luckyluckster lol so we are quoting tupac now? i am not mad at it	0.21758725	1	*Confusing*
@hye_jin i'm actually back at tcnj taking classes omgsh you're in belize?! so lucky!! how long are you there for?? oo what are the NCLEX?	0.5652111	0	*Looks mislabeled*
@v18rocks MAGIC. only hope left.	0.21589482	1	*Looks mislabeled*

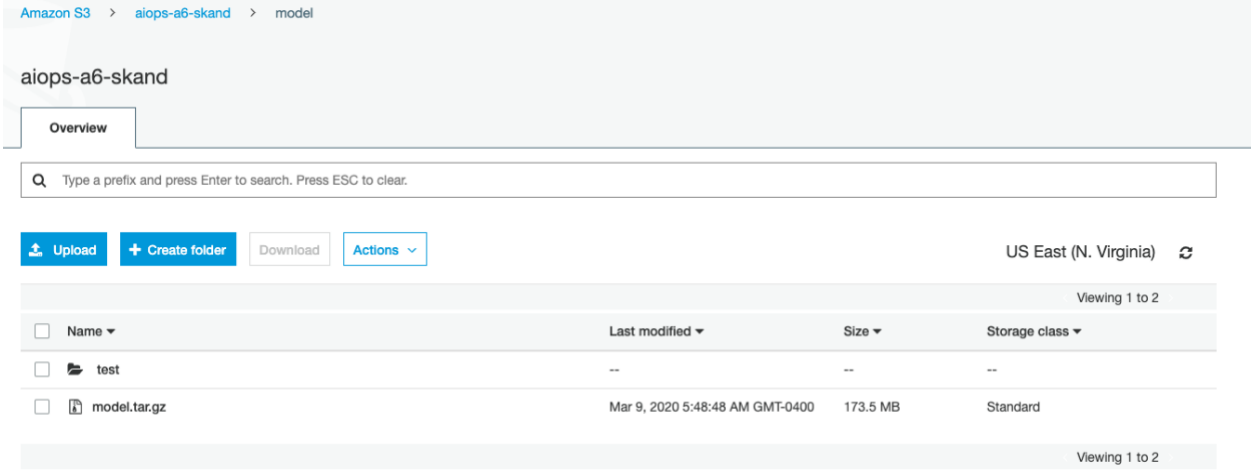
Most of the misclassification that we found while skimming at the misclassified tweets were mostly correct or just could not be improved any further. 20% of error looks like a reasonable estimate that even a human would be prone to while labelling the tweets. Therefore, we believe that the model performs as per our expectations.

Step 6: Sagemaker Implementation for hosting model:

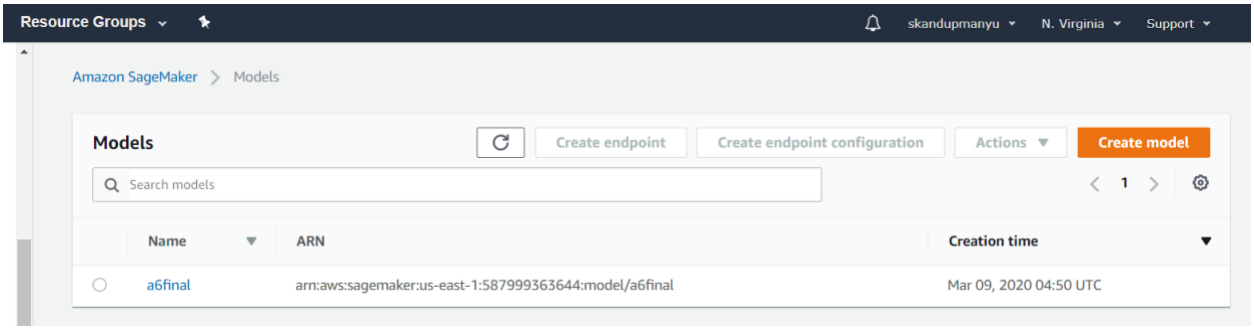
Model implementation required serving the model as an endpoint. Our models with 200D embeddings were huge in size (>3GB) and AWS Educate as well as free tier account did not allow us to serve these models.

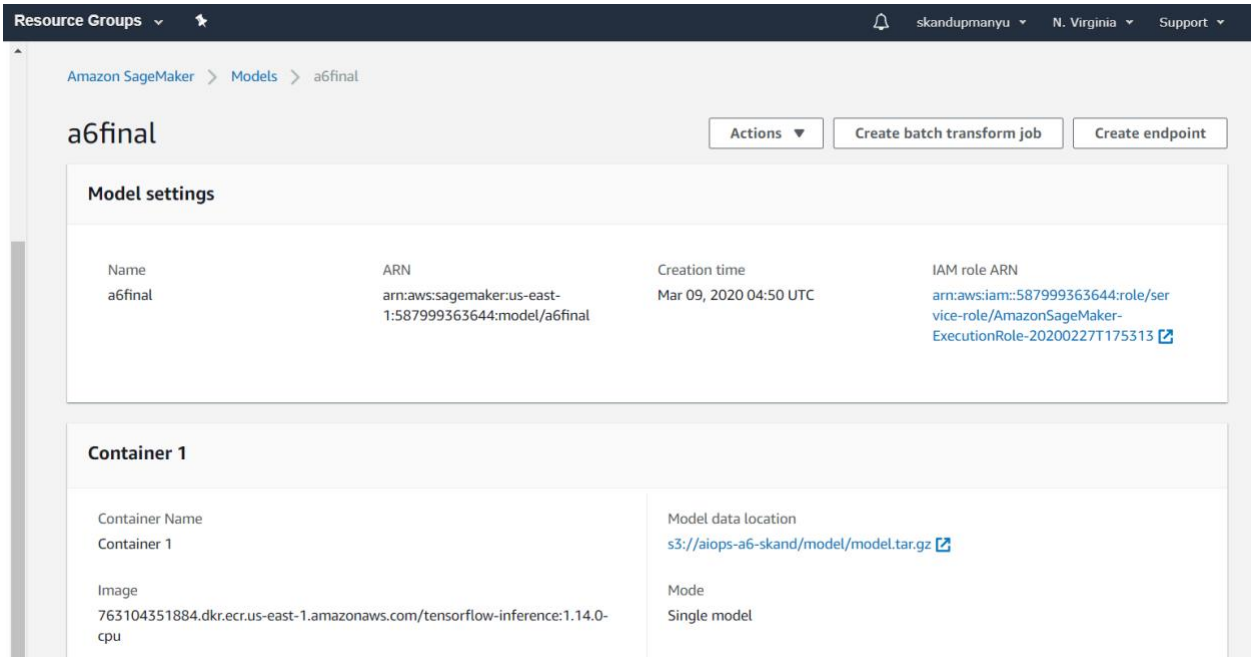
To reduce the model size we kept out model architecture the same but reduced our embedding dimensions to 25D. However, this led to a very small decrease in test accuracy (~0.25%)

Stored model in S3:

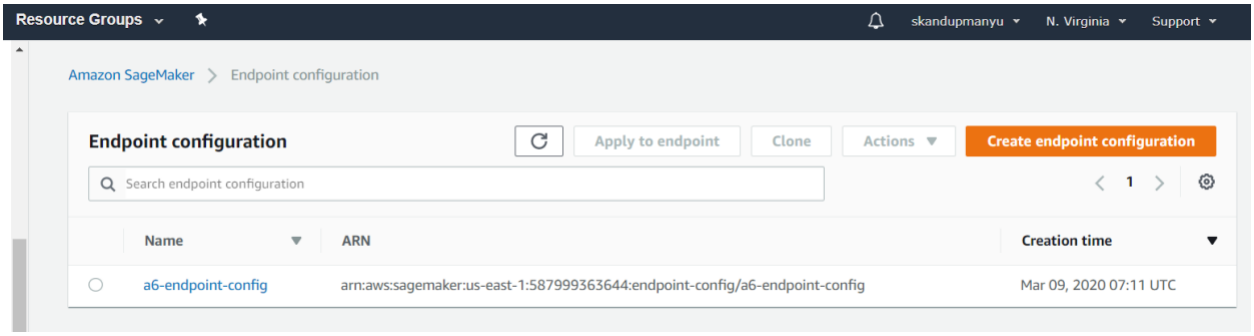


Sagemaker Model:

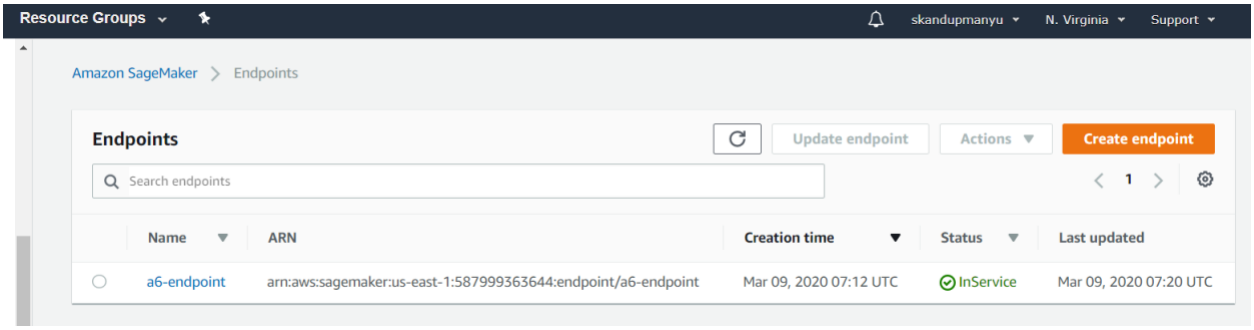


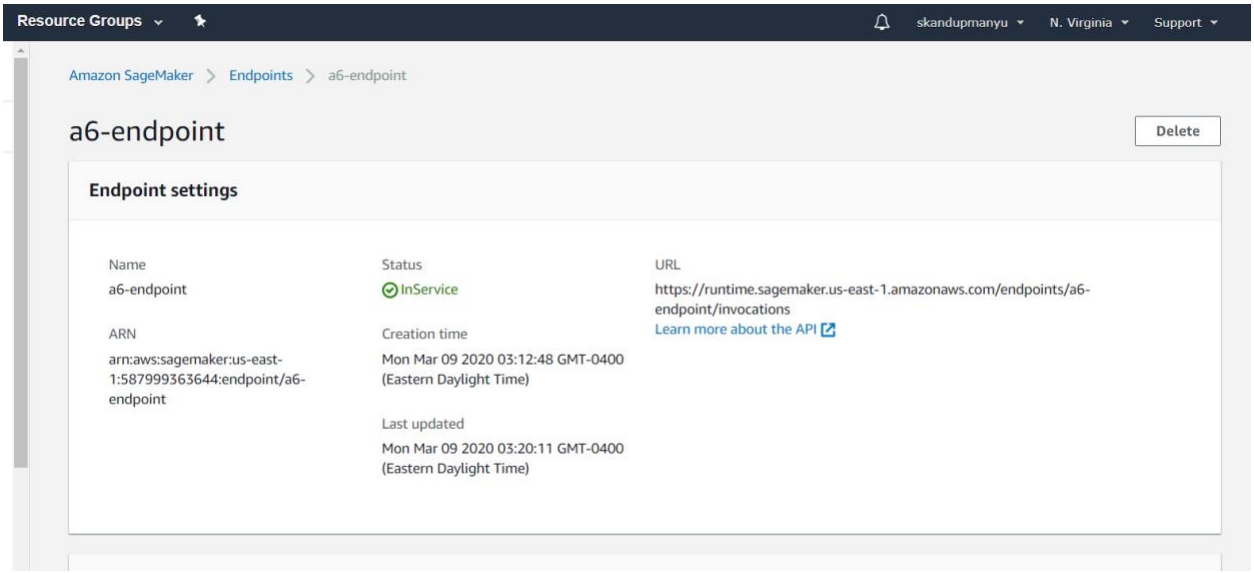


Config:



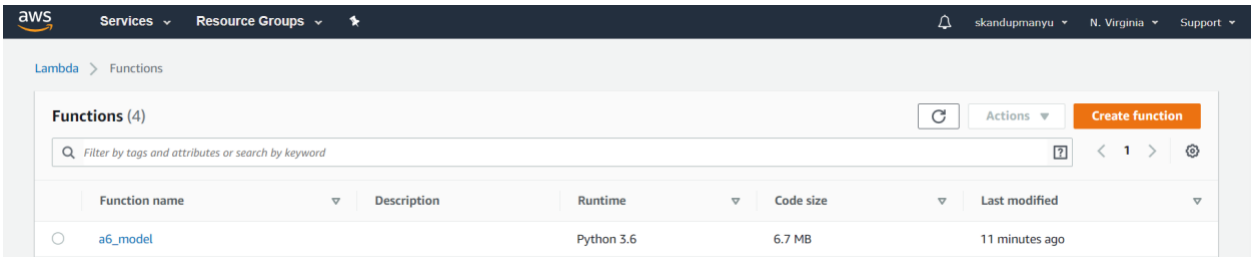
Endpoint:

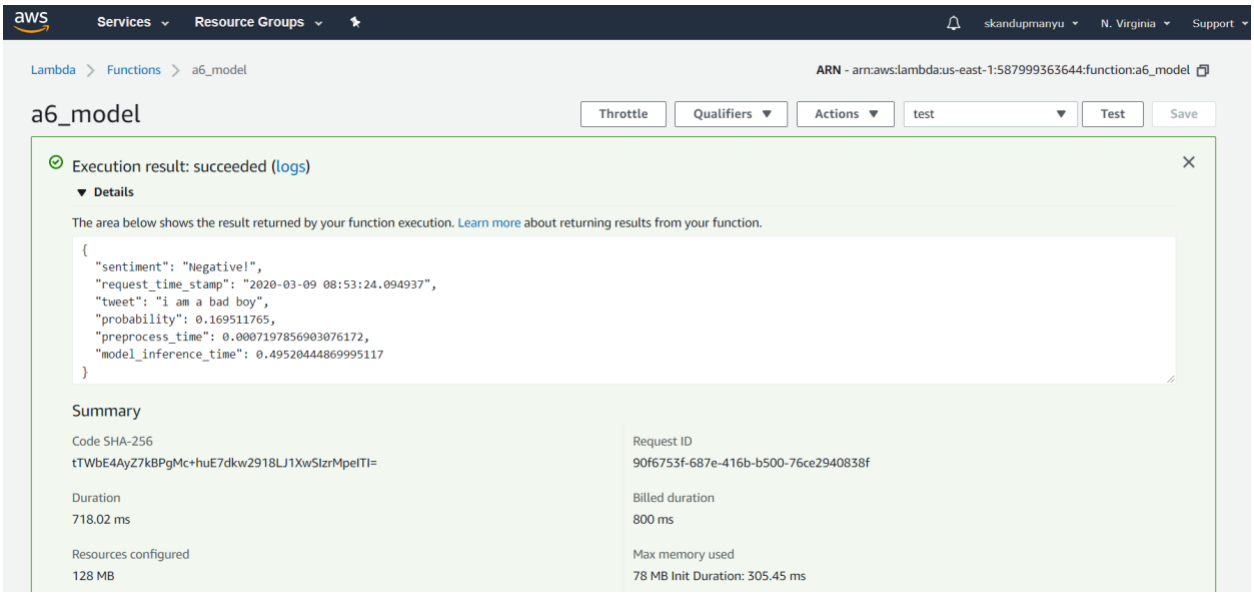




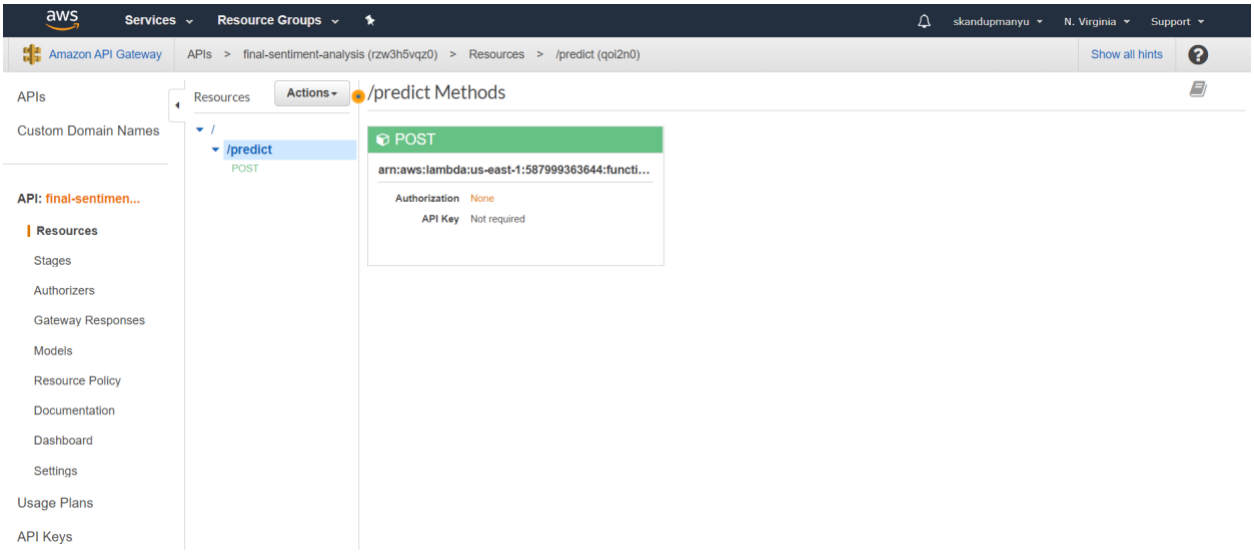
Step 7: API Gateway and Lambda implementation:

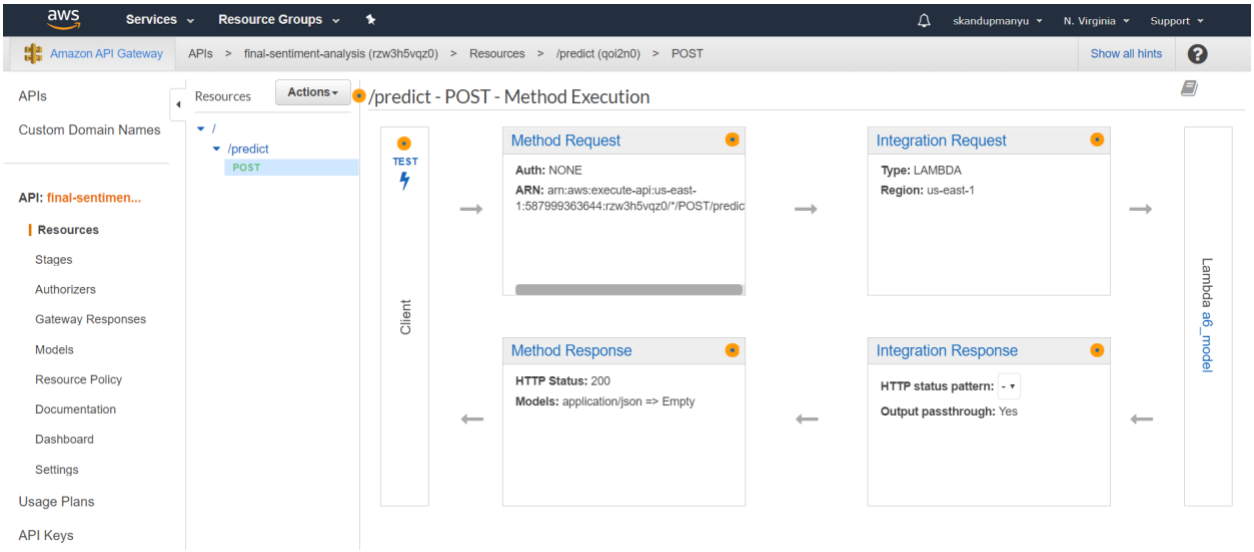
Lambda:

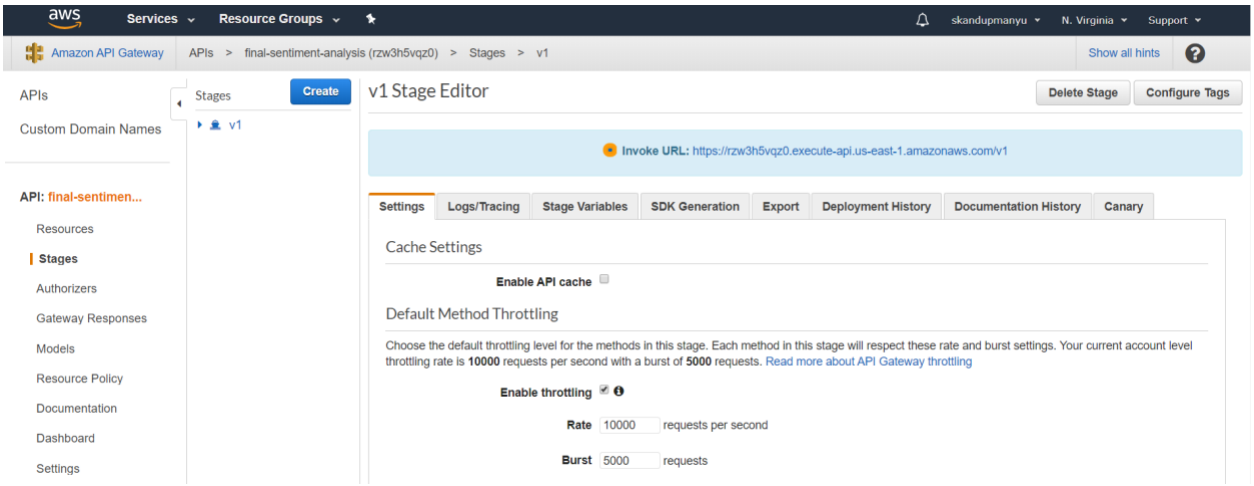


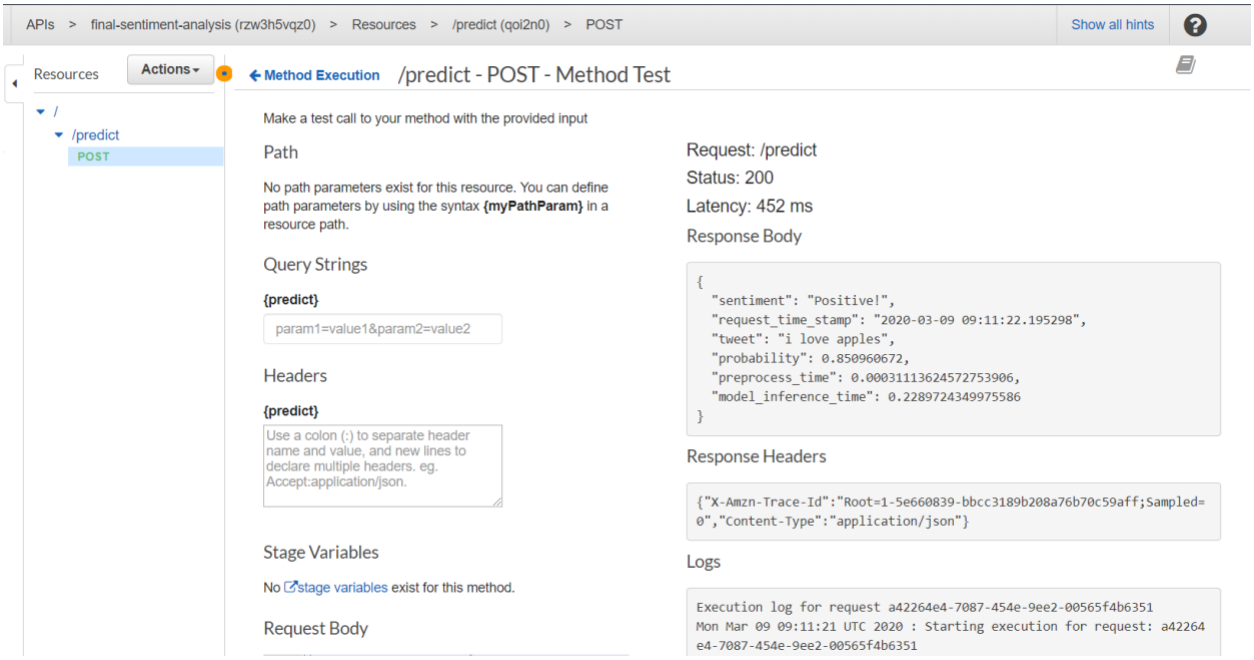


API Gateway:




aws
Services
Resource Groups
skandupmanyu
N. Virginia
Support
Amazon API Gateway
APIs > final-sentiment-analysis (rzw3h5vqz0) > Resources > /predict (qoi2n0) > POST
Show all hints
APIs
Custom Domain Names
API: final-sentimen...
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Dashboard
Settings
Usage Plans
API Keys
Resources
Actions
/
/predict
POST
/predict - POST - Method Execution
TEST
Client
Method Request
Auth: NONE
ARN: arn:aws:execute-api:us-east-1:587999363644:rzw3h5vqz0/*/POST/predic
Integration Request
Type: LAMBDA
Region: us-east-1
Lambda a6_model
Method Response
HTTP Status: 200
Models: application/json => Empty
Integration Response
HTTP status pattern: -
Output passthrough: Yes

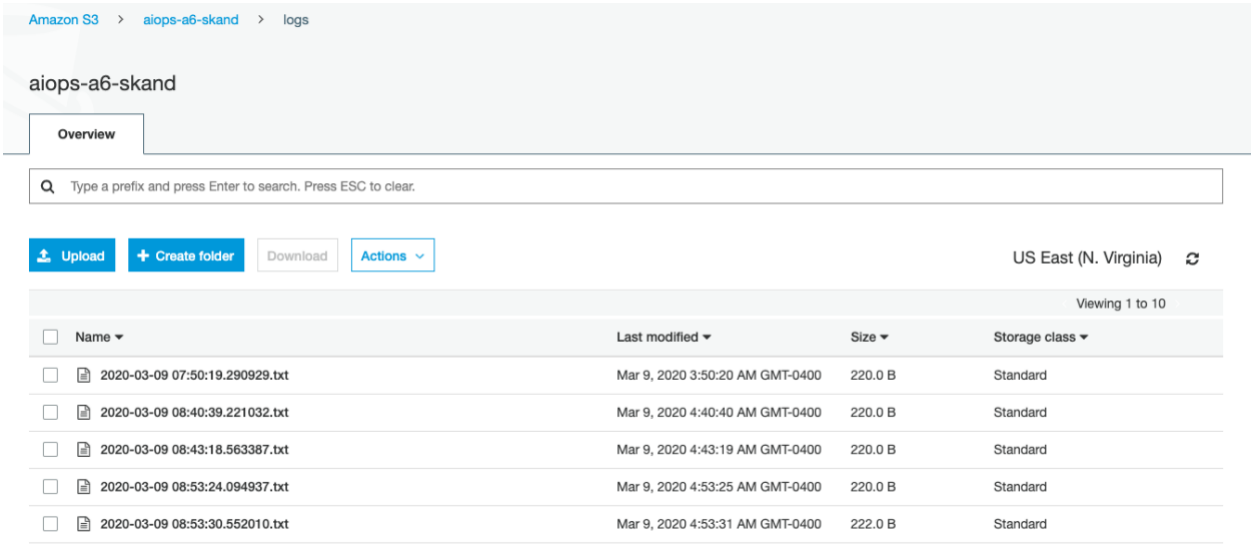

aws
Services
Resource Groups
skandupmanyu
N. Virginia
Support
Amazon API Gateway
APIs > final-sentiment-analysis (rzw3h5vqz0) > Stages > v1
Show all hints
APIs
Custom Domain Names
API: final-sentimen...
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Dashboard
Settings
Stages
Create
v1
v1 Stage Editor
Delete Stage
Configure Tags
Invoke URL: https://rzw3h5vqz0.execute-api.us-east-1.amazonaws.com/v1
Settings
Logs/Tracing
Stage Variables
SDK Generation
Export
Deployment History
Documentation History
Canary
Cache Settings
Enable API cache
Default Method Throttling
Choose the default throttling level for the methods in this stage. Each method in this stage will respect these rate and burst settings. Your current account level throttling rate is 10000 requests per second with a burst of 5000 requests. Read more about API Gateway throttling
Enable throttling
Rate 10000 requests per second
Burst 5000 requests



API call for the final model:

```
(base) Skands-MacBook-Air:~ skand$ curl -X POST https://rzw3h5vqz0.execute-api.us-east-1.amazonaws.com/v1/predict --header "Content-Type:application/json" --data '{"tweet": "I love my life"}'
{"sentiment": "Positive!", "request_time_stamp": "2020-03-09 17:13:07.266727", "tweet": "I love my life", "probability": 0.848078251, "preprocess_time": 0.00012183189392089844, "model_inference_time": 0.04193758964538574}(base) Skands-MacBook-Air:~ skand$
```

Logs:



API Gateway link

https://rzw3h5vqz0.execute-api.us-east-1.amazonaws.com/v1/predict

API call:

curl -X POST https://rzw3h5vqz0.execute-api.us-east-1.amazonaws.com/v1/predict --header "Content-Type:application/json" --data '{"tweet": "I love my life"}'

Github:

https://github.com/skandupmanyu/AI-Ops-A6

Notes:

- If you find the endpoint is not active, kindly let us know. We'll start it immediately.
- Also, kindly let us know when the evaluation is complete, so that we can delete our model endpoint (our model is being served in our private account due to limitations in AWS Educate)