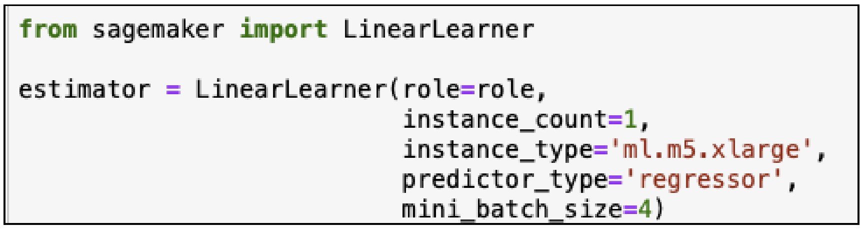
Ay 2ML Modeling with Amazon SageMaker
Mico Ellerich Comia
May 15, 2021

- SageMaker (Lesson 1)
    - Overview
        - End-to-end solution for building, training, and deploying ML models quickly.
        - Refer to https://aws.amazon.com/blogs/machine-learning/category/artificial-intelligence/sagemaker/ for sample implementations.
    - Training
        - Usage of built-in algorithms (LinearLearner, XGBoost, etc.)
        - Capable of manual or automated hyperparameter tuning
        - Uses a separate training instance
    - Deployment and evaluation
        - Realtime Inference
            - Helps with quick deployment
            - Can be thought of as a web application where you send your prediction request
        - Batch inference
            - Input data is stored in S3
            - Used when you have many records. Not advisable if you only have 1 record.
    - Managed Notebook Instances
        - Isolated environment
        - Requires:
            - Notebook instance name
            - Notebook instance type (changeable)
        - Environment:
            - Jupyter Notebook
            - Python3 kernel
            - MXNet kernel
            - PyTorch kernel

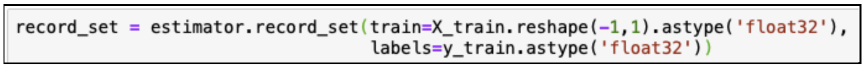
- SageMaker SDK / Basic Usage (Lesson 2)
    - SageMaker Python SDK
        - Abstraction layer that is specific to a language
        - Can be installed in local machines
        - A script that uses SDK automatically calls API
    - Basic Usage

1. Estimator
    - Prepare or initialize configuration for estimation job
    - Includes execution role, instance count and type, etc.
    - Where users can define hyperparameter values for chosen algorithm.



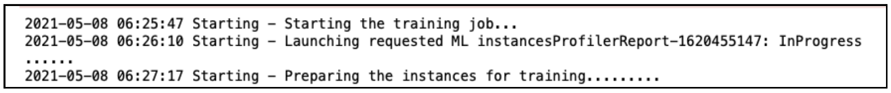
```
from sagemaker import LinearLearner

estimator = LinearLearner(role=role,
                          instance_count=1,
                          instance_type='ml.m5.xlarge',
                          predictor_type='regressor',
                          mini_batch_size=4)
```

    - "record_set" creates a record object that is automatically stored in S3.



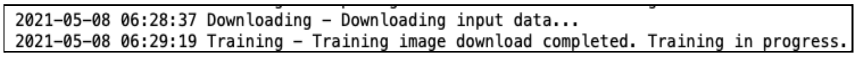
```
record_set = estimator.record_set(train=X_train.reshape(-1,1).astype('float32'),
                                   labels=y_train.astype('float32'))
```

2. Fit
    - Similar across different algorithms.
    - ML instances are prepared by creating a separate instance for training



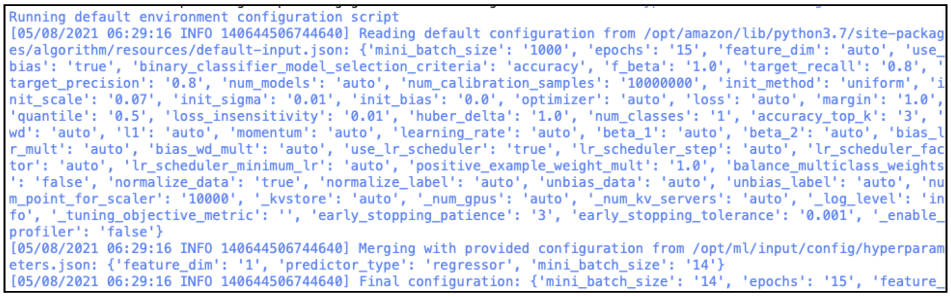
```
2021-05-08 06:25:47 Starting - Starting the training job...
2021-05-08 06:26:10 Starting - Launching requested ML instancesProfilerReport-1620455147: InProgress
......
2021-05-08 06:27:17 Starting - Preparing the instances for training.........
```

    - Downloads input data (training data stored in S3)



```
2021-05-08 06:28:37 Downloading - Downloading input data...
2021-05-08 06:29:19 Training - Training image download completed. Training in progress.
```

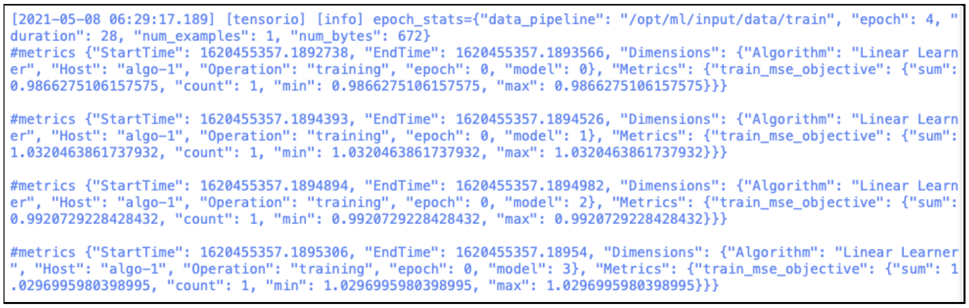
    - Algorithms use Docker container images that have default hyperparameter values. User-set hyperparameter values are merged with default values to create the final configuration.



```
Running default environment configuration script
[05/08/2021 06:29:16 INFO 140644506744640] Reading default configuration from /opt/amazon/lib/python3.7/site-packag
es/algorithm/resources/default-input.json: {'mini_batch_size': '1000', 'epochs': '15', 'feature_dim': 'auto', 'use_
bias': 'true', 'binary_classifier_model_selection_criteria': 'accuracy', 'f_beta': '1.0', 'target_recall': '0.8', '
target_precision': '0.8', 'num_models': 'auto', 'num_calibration_samples': '10000000', 'init_method': 'uniform', 'i
nit_scale': '0.07', 'init_sigma': '0.01', 'init_bias': '0.0', 'optimizer': 'auto', 'loss': 'auto', 'margin': '1.0',
'quantile': '0.5', 'loss_insensitivity': '0.01', 'huber_delta': '1.0', 'num_classes': '1', 'accuracy_top_k': '3', '
wd': 'auto', 'l1': 'auto', 'momentum': 'auto', 'learning_rate': 'auto', 'beta_1': 'auto', 'beta_2': 'auto', 'bias_l
r_mult': 'auto', 'bias_wd_mult': 'auto', 'use_lr_scheduler': 'true', 'lr_scheduler_step': 'auto', 'lr_scheduler_fac
tor': 'auto', 'lr_scheduler_minimum_lr': 'auto', 'positive_example_weight_mult': '1.0', 'balance_multiclass_weights
': 'false', 'normalize_data': 'true', 'normalize_label': 'auto', 'unbias_data': 'auto', 'unbias_label': 'auto', 'nu
m_point_for_scaler': '10000', '_kvstore': 'auto', '_num_gpus': 'auto', '_num_kv_servers': 'auto', '_log_level': 'in
fo', '_tuning_objective_metric': '', 'early_stopping_patience': '3', 'early_stopping_tolerance': '0.001', '_enable_
profiler': 'false'}
[05/08/2021 06:29:16 INFO 140644506744640] Merging with provided configuration from /opt/ml/input/config/hyperparam
eters.json: {'feature_dim': '1', 'predictor_type': 'regressor', 'mini_batch_size': '14'}
[05/08/2021 06:29:16 INFO 140644506744640] Final configuration: {'mini_batch_size': '14', 'epochs': '15', 'feature_
```

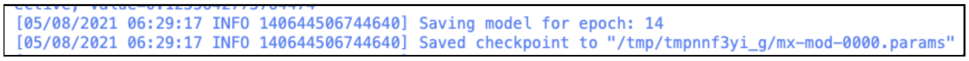
    - Container contains scripts that trigger specific tasks, in this case, it triggers the training script. Most of the logs are from the training script.
    - Training script contains metrics and finds the best model recursively. Can set early stopping if the model is not improving

after multiple epochs.



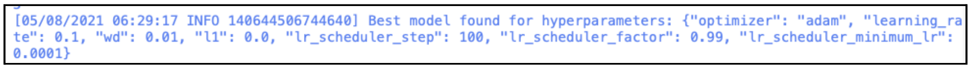
```
[2021-05-08 06:29:17.189] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/train", "epoch": 4, "
duration": 28, "num_examples": 1, "num_bytes": 672}
#metrics {"StartTime": 1620455357.1892738, "EndTime": 1620455357.1893566, "Dimensions": {"Algorithm": "Linear Learn
er", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 0}, "Metrics": {"train_mse_objective": {"sum":
0.9866275106157575, "count": 1, "min": 0.9866275106157575, "max": 0.9866275106157575}}}

#metrics {"StartTime": 1620455357.1894393, "EndTime": 1620455357.1894526, "Dimensions": {"Algorithm": "Linear Learn
er", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 1}, "Metrics": {"train_mse_objective": {"sum":
1.0320463861737932, "count": 1, "min": 1.0320463861737932, "max": 1.0320463861737932}}}

#metrics {"StartTime": 1620455357.1894894, "EndTime": 1620455357.1894982, "Dimensions": {"Algorithm": "Linear Learn
er", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 2}, "Metrics": {"train_mse_objective": {"sum":
0.9920729228428432, "count": 1, "min": 0.9920729228428432, "max": 0.9920729228428432}}}

#metrics {"StartTime": 1620455357.1895306, "EndTime": 1620455357.18954, "Dimensions": {"Algorithm": "Linear Learner
", "Host": "algo-1", "Operation": "training", "epoch": 0, "model": 3}, "Metrics": {"train_mse_objective": {"sum": 1
.0296995980398995, "count": 1, "min": 1.0296995980398995, "max": 1.0296995980398995}}}
```

- Saves model of each epoch.



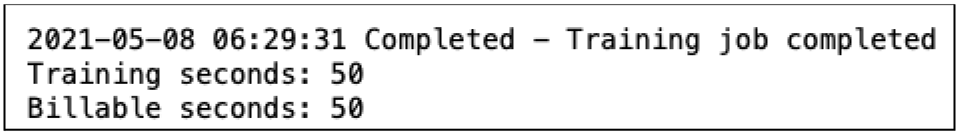
```
[05/08/2021 06:29:17 INFO 140644506744640] Saving model for epoch: 14
[05/08/2021 06:29:17 INFO 140644506744640] Saved checkpoint to "/tmp/tmpnnf3yi_g/mx-mod-0000.params"
```

- Provides the hyperparameters of the best model found.



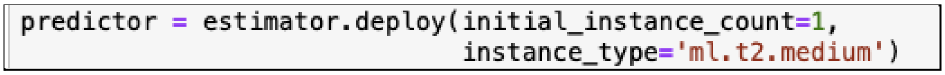
```
[05/08/2021 06:29:17 INFO 140644506744640] Best model found for hyperparameters: {"optimizer": "adam", "learning_ra
te": 0.1, "wd": 0.01, "l1": 0.0, "lr_scheduler_step": 100, "lr_scheduler_factor": 0.99, "lr_scheduler_minimum_lr":
0.0001}
```

- If validation data is not provided, “test data is not provided” will show up.
- Returns training seconds (how long the training job ran).



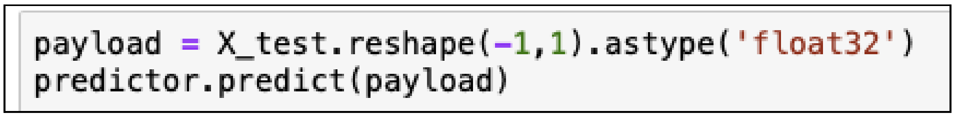
```
2021-05-08 06:29:31 Completed - Training job completed
Training seconds: 50
Billable seconds: 50
```

3. Deploy
    - Creates a “prediction endpoint” instance or separate instance for prediction.



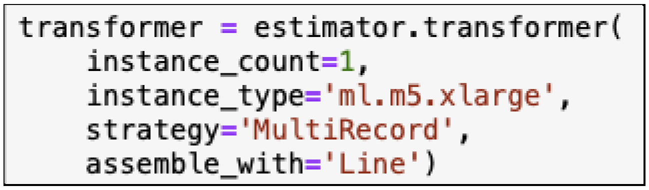
```
predictor = estimator.deploy(initial_instance_count=1,
                             instance_type='ml.t2.medium')
```

- “.predict” passes a payload for prediction.



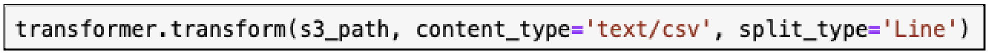
```
payload = X_test.reshape(-1,1).astype('float32')
predictor.predict(payload)
```

4. Transform
    - Does not need an endpoint to perform predictions.
    - Estimator also sets configuration for estimator job.



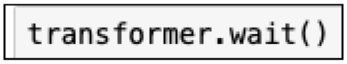
```
transformer = estimator.transformer(
    instance_count=1,
    instance_type='ml.m5.xlarge',
    strategy='MultiRecord',
    assemble_with='Line')
```

- “.transform” creates a transform job and gets input data from the specified S3 path. Creates an S3 output path and similar to “.fit”

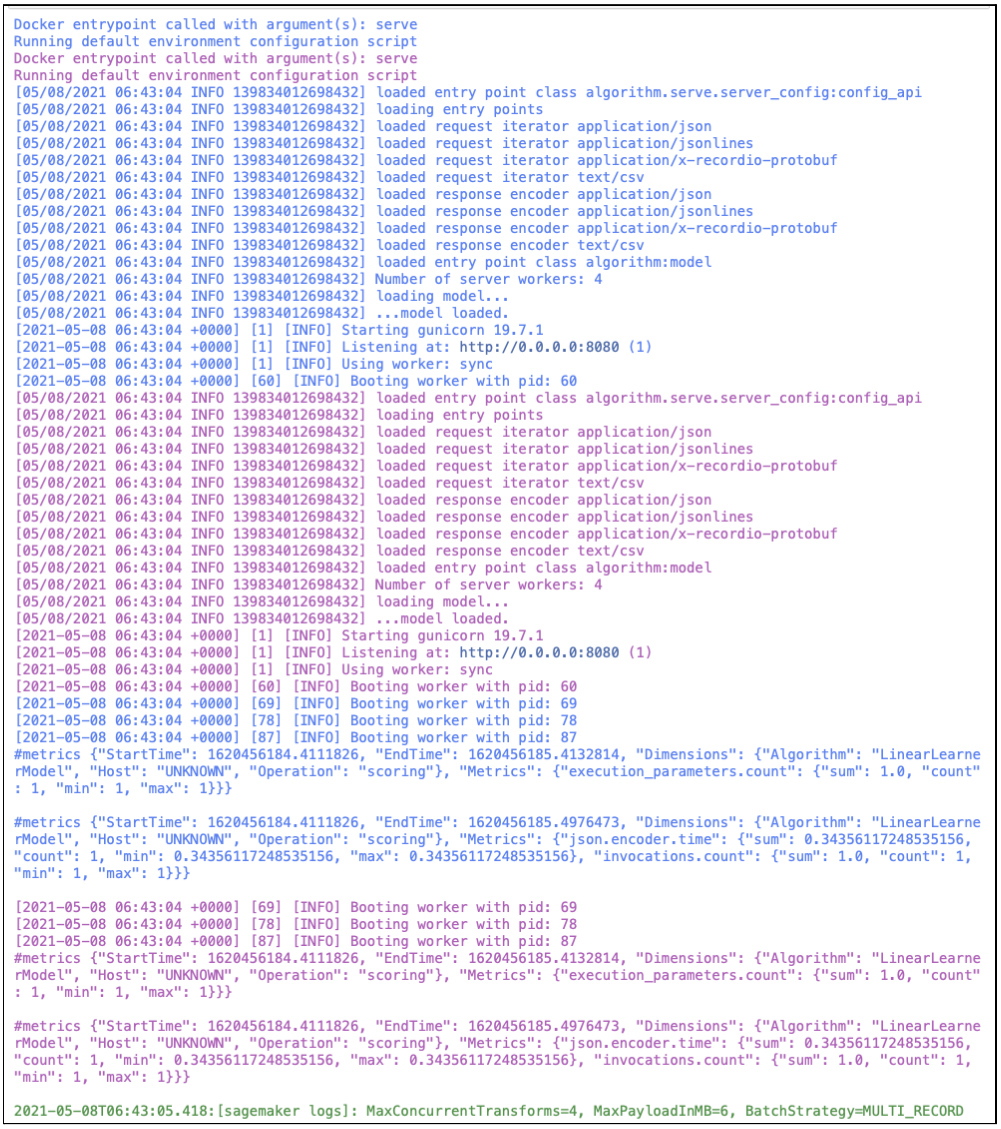


```
transformer.transform(s3_path, content_type='text/csv', split_type='Line')
```

- “.wait” attribute waits until the transformer job is complete. Default is true. “.wait()”method is a placeholder docstring that shows the logs of the transformer.



```
transformer.wait()
```



```
Docker entrypoint called with argument(s): serve
Running default environment configuration script
Docker entrypoint called with argument(s): serve
Running default environment configuration script
[05/08/2021 06:43:04 INFO 139834012698432] loaded entry point class algorithm.serve.server_config:config_api
[05/08/2021 06:43:04 INFO 139834012698432] loading entry points
[05/08/2021 06:43:04 INFO 139834012698432] loaded request iterator application/json
[05/08/2021 06:43:04 INFO 139834012698432] loaded request iterator application/jsonlines
[05/08/2021 06:43:04 INFO 139834012698432] loaded request iterator application/x-recordio-protobuf
[05/08/2021 06:43:04 INFO 139834012698432] loaded request iterator text/csv
[05/08/2021 06:43:04 INFO 139834012698432] loaded response encoder application/json
[05/08/2021 06:43:04 INFO 139834012698432] loaded response encoder application/jsonlines
[05/08/2021 06:43:04 INFO 139834012698432] loaded response encoder application/x-recordio-protobuf
[05/08/2021 06:43:04 INFO 139834012698432] loaded response encoder text/csv
[05/08/2021 06:43:04 INFO 139834012698432] loaded entry point class algorithm:model
[05/08/2021 06:43:04 INFO 139834012698432] Number of server workers: 4
[05/08/2021 06:43:04 INFO 139834012698432] loading model...
[05/08/2021 06:43:04 INFO 139834012698432] ...model loaded.
[2021-05-08 06:43:04 +0000] [1] [INFO] Starting gunicorn 19.7.1
[2021-05-08 06:43:04 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080 (1)
[2021-05-08 06:43:04 +0000] [1] [INFO] Using worker: sync
[2021-05-08 06:43:04 +0000] [60] [INFO] Booting worker with pid: 60
[05/08/2021 06:43:04 INFO 139834012698432] loaded entry point class algorithm.serve.server_config:config_api
[05/08/2021 06:43:04 INFO 139834012698432] loading entry points
[05/08/2021 06:43:04 INFO 139834012698432] loaded request iterator application/json
[05/08/2021 06:43:04 INFO 139834012698432] loaded request iterator application/jsonlines
[05/08/2021 06:43:04 INFO 139834012698432] loaded request iterator application/x-recordio-protobuf
[05/08/2021 06:43:04 INFO 139834012698432] loaded request iterator text/csv
[05/08/2021 06:43:04 INFO 139834012698432] loaded response encoder application/json
[05/08/2021 06:43:04 INFO 139834012698432] loaded response encoder application/jsonlines
[05/08/2021 06:43:04 INFO 139834012698432] loaded response encoder application/x-recordio-protobuf
[05/08/2021 06:43:04 INFO 139834012698432] loaded response encoder text/csv
[05/08/2021 06:43:04 INFO 139834012698432] loaded entry point class algorithm:model
[05/08/2021 06:43:04 INFO 139834012698432] Number of server workers: 4
[05/08/2021 06:43:04 INFO 139834012698432] loading model...
[05/08/2021 06:43:04 INFO 139834012698432] ...model loaded.
[2021-05-08 06:43:04 +0000] [1] [INFO] Starting gunicorn 19.7.1
[2021-05-08 06:43:04 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080 (1)
[2021-05-08 06:43:04 +0000] [1] [INFO] Using worker: sync
[2021-05-08 06:43:04 +0000] [60] [INFO] Booting worker with pid: 60
[2021-05-08 06:43:04 +0000] [69] [INFO] Booting worker with pid: 69
[2021-05-08 06:43:04 +0000] [78] [INFO] Booting worker with pid: 78
[2021-05-08 06:43:04 +0000] [87] [INFO] Booting worker with pid: 87
#metrics {"StartTime": 1620456184.4111826, "EndTime": 1620456185.4132814, "Dimensions": {"Algorithm": "LinearLearne
rModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0, "count"
: 1, "min": 1, "max": 1}}}

#metrics {"StartTime": 1620456184.4111826, "EndTime": 1620456185.4976473, "Dimensions": {"Algorithm": "LinearLearne
rModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 0.34356117248535156,
"count": 1, "min": 0.34356117248535156, "max": 0.34356117248535156}, "invocations.count": {"sum": 1.0, "count": 1,
"min": 1, "max": 1}}}

[2021-05-08 06:43:04 +0000] [69] [INFO] Booting worker with pid: 69
[2021-05-08 06:43:04 +0000] [78] [INFO] Booting worker with pid: 78
[2021-05-08 06:43:04 +0000] [87] [INFO] Booting worker with pid: 87
#metrics {"StartTime": 1620456184.4111826, "EndTime": 1620456185.4132814, "Dimensions": {"Algorithm": "LinearLearne
rModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"execution_parameters.count": {"sum": 1.0, "count"
: 1, "min": 1, "max": 1}}}

#metrics {"StartTime": 1620456184.4111826, "EndTime": 1620456185.4976473, "Dimensions": {"Algorithm": "LinearLearne
rModel", "Host": "UNKNOWN", "Operation": "scoring"}, "Metrics": {"json.encoder.time": {"sum": 0.34356117248535156,
"count": 1, "min": 0.34356117248535156, "max": 0.34356117248535156}, "invocations.count": {"sum": 1.0, "count": 1,
"min": 1, "max": 1}}}

2021-05-08T06:43:05.418:[sagemaker logs]: MaxConcurrentTransforms=4, MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
```