XGBoost Built-in Algorithm - Bike Rental Regression Example

In [1]:
```
import numpy as np
import pandas as pd

import boto3
import re

import sagemaker
from sagemaker import get_execution_role
# SageMaker SDK Documentation: http://sagemaker.readthedocs.io/en/latest/estimators.html
```

Upload Data to S3

In [2]:
```
# Specify your bucket name
bucket_name = 'dwb-ml-sagemaker'

training_folder = r'bikerental/training/'
validation_folder = r'bikerental/validation/'
test_folder = r'bikerental/test/'

s3_model_output_location = r's3://{0}/bikerental/model'.format(bucket_name)
s3_training_file_location = r's3://{0}/{1}'.format(bucket_name,training_folder)
s3_validation_file_location = r's3://{0}/{1}'.format(bucket_name,validation_folder)
s3_test_file_location = r's3://{0}/{1}'.format(bucket_name,test_folder)
```

In [3]:
```
print(s3_model_output_location)
print(s3_training_file_location)
print(s3_validation_file_location)
print(s3_test_file_location)
```

```
s3://dwb-ml-sagemaker/bikerental/model
s3://dwb-ml-sagemaker/bikerental/training/
s3://dwb-ml-sagemaker/bikerental/validation/
s3://dwb-ml-sagemaker/bikerental/test/
```

In [4]:
```
# Write and Reading from S3 is just as easy
# files are referred as objects in S3.
# file name is referred as key name in S3

# File stored in S3 is automatically replicated across 3 different availability zones
# in the region where the bucket was created.

# http://boto3.readthedocs.io/en/latest/guide/s3.html
def write_to_s3(filename, bucket, key):
    with open(filename,'rb') as f: # Read in binary mode
        return boto3.Session().resource('s3').Bucket(bucket).Object(key).upload_fileobj(f)
```

In [5]:
```
write_to_s3('bike_train.csv',
            bucket_name,
            training_folder + 'bike_train.csv')

write_to_s3('bike_validation.csv',
            bucket_name,
            validation_folder + 'bike_validation.csv')

write_to_s3('bike_test.csv',
            bucket_name,
            test_folder + 'bike_test.csv')
```

Training Algorithm Docker Image

SageMaker maintains a separate image for algorithm and region

https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-algo-docker-registry-paths.html

In [6]:
```
# Use Spot Instance - Save up to 90% of training cost by using spot instances when compared to on-demand instances
# Reference: https://github.com/aws-samples/amazon-sagemaker-managed-spot-training/blob/main/xgboost_built_in_managed

# if you are still on two-month free-tier you can use the on-demand instance by setting:
#   use_spot_instances = False

# We will use spot for training
#DWB 2023-07-27, still in free tier#use_spot_instances = True
```

```
use_spot_instances = False
max_run = 3600 # in seconds
max_wait = 7200 if use_spot_instances else None # in seconds

job_name = 'xgboost-bikerental-v1'

checkpoint_s3_uri = None

if use_spot_instances:
    checkpoint_s3_uri = f's3://{bucket_name}/bikerental/checkpoints/{job_name}'

print (f'Checkpoint uri: {checkpoint_s3_uri}')
```

```
Checkpoint uri: None
```

In [7]:
```
# Establish a session with AWS
sess = sagemaker.Session()
```

In [8]:
```
role = get_execution_role()
```

In [9]:
```
# This role contains the permissions needed to train, deploy models
# SageMaker Service is trusted to assume this role
print(role)
```

```
arn:aws:iam::030021571292:role/service-role/AmazonSageMaker-ExecutionRole-20230702T224355
```

In [10]:
```
# https://sagemaker.readthedocs.io/en/stable/api/utility/image_uris.html#sagemaker.image_uris.retrieve

# SDK 2 uses image_uris.retrieve the container image location

# Use XGBoost 1.2 version
container = sagemaker.image_uris.retrieve("xgboost",sess.boto_region_name,version="1.2-2")

print (f'Using XGBoost Container {container}')
```

```
Using XGBoost Container 683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.2-2
```

Build Model

In [13]:
```
# Configure the training job
# Specify type and number of instances to use
```

```
# S3 location where final artifacts needs to be stored

#   Reference: http://sagemaker.readthedocs.io/en/latest/estimators.html

# for managed spot training, specify the use_spot_instances flag, max_run, max_wait and checkpoint_s3_uri

# SDK 2.x version does not require train prefix for instance count and type
estimator = sagemaker.estimator.Estimator(
    container,
    role,
    instance_count=1,
    instance_type='ml.m5.xlarge',
    output_path=s3_model_output_location,
    sagemaker_session=sess,
    base_job_name = job_name,
    use_spot_instances=use_spot_instances,
    max_run=max_run,
    max_wait=max_wait,
    checkpoint_s3_uri=checkpoint_s3_uri)
```

In [14]:
```
# Specify hyper parameters that appropriate for the training algorithm
# XGBoost Training Parameter Reference
#  https://github.com/dmlc/xgboost/blob/master/doc/parameter.rst#learning-task-parameters
estimator.set_hyperparameters(max_depth=5,
                              objective="reg:squarederror",
                              eta=0.1,
                              num_round=150)
```

In [15]:
```
estimator.hyperparameters()
```

Out[15]:
```
{'max_depth': 5, 'objective': 'reg:squarederror', 'eta': 0.1, 'num_round': 150}
```

Specify Training Data Location and Optionally, Validation Data Location

In [16]:
```
# content type can be libsvm or csv for XGBoost
training_input_config = sagemaker.session.TrainingInput(
    s3_data=s3_training_file_location,
    content_type='csv',
    s3_data_type='S3Prefix')
```

```
validation_input_config = sagemaker.session.TrainingInput(
    s3_data=s3_validation_file_location,
    content_type='csv',
    s3_data_type='S3Prefix'
)

data_channels = {'train': training_input_config, 'validation': validation_input_config}
```

In [17]:
```
print(training_input_config.config)
print(validation_input_config.config)
```

```
{'DataSource': {'S3DataSource': {'S3DataType': 'S3Prefix', 'S3Uri': 's3://dwb-ml-sagemaker/bikerental/training/', 'S
3DataDistributionType': 'FullyReplicated'}}, 'ContentType': 'csv'}
{'DataSource': {'S3DataSource': {'S3DataType': 'S3Prefix', 'S3Uri': 's3://dwb-ml-sagemaker/bikerental/validation/',
'S3DataDistributionType': 'FullyReplicated'}}, 'ContentType': 'csv'}
```

Train the model

In [18]:
```
# XGBoost supports "train", "validation" channels
# Reference: Supported channels by algorithm
#   https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-algo-docker-registry-paths.html
estimator.fit(data_channels)
```

```
INFO:sagemaker:Creating training-job with name: xgboost-bikerental-v1-2023-07-27-21-46-08-347
```

```
2023-07-27 21:46:08 Starting - Starting the training job...
2023-07-27 21:46:23 Starting - Preparing the instances for training......
2023-07-27 21:47:39 Downloading - Downloading input data...
2023-07-27 21:47:59 Training - Downloading the training image......
2023-07-27 21:49:00 Uploading - Uploading generated training model[2023-07-27 21:48:55.417 ip-10-2-99-157.ec2.intern
al:7 INFO utils.py:27] RULE_JOB_STOP_SIGNAL_FILENAME: None
[2023-07-27:21:48:55:INFO] Imported framework sagemaker_xgboost_container.training
[2023-07-27:21:48:55:INFO] Failed to parse hyperparameter objective value reg:squarederror to Json.
Returning the value itself
[2023-07-27:21:48:55:INFO] No GPUs detected (normal if no gpus installed)
[2023-07-27:21:48:55:INFO] Running XGBoost Sagemaker in algorithm mode
[2023-07-27:21:48:55:INFO] Determined delimiter of CSV input is ','
[2023-07-27:21:48:55:INFO] Determined delimiter of CSV input is ','
[2023-07-27:21:48:55:INFO] Determined delimiter of CSV input is ','
[2023-07-27:21:48:55:INFO] Determined delimiter of CSV input is ','
[2023-07-27:21:48:55:INFO] Single node training.
[2023-07-27:21:48:55:INFO] Train matrix has 7620 rows and 13 columns
[2023-07-27:21:48:55:INFO] Validation matrix has 3266 rows
[2023-07-27 21:48:55.502 ip-10-2-99-157.ec2.internal:7 INFO json_config.py:91] Creating hook from json_config at /op
t/ml/input/config/debughookconfig.json.
[2023-07-27 21:48:55.503 ip-10-2-99-157.ec2.internal:7 INFO hook.py:201] tensorboard_dir has not been set for the ho
ok. SMDebug will not be exporting tensorboard summaries.
[2023-07-27 21:48:55.503 ip-10-2-99-157.ec2.internal:7 INFO profiler_config_parser.py:102] User has disabled profile
r.
[2023-07-27 21:48:55.504 ip-10-2-99-157.ec2.internal:7 INFO hook.py:255] Saving to /opt/ml/output/tensors
[2023-07-27 21:48:55.504 ip-10-2-99-157.ec2.internal:7 INFO state_store.py:77] The checkpoint config file /opt/ml/in
put/config/checkpointconfig.json does not exist.
[2023-07-27:21:48:55:INFO] Debug hook created from config
[0]#011train-rmse:3.90481#011validation-rmse:3.91500
[2023-07-27 21:48:55.511 ip-10-2-99-157.ec2.internal:7 INFO hook.py:423] Monitoring the collections: metrics
[2023-07-27 21:48:55.513 ip-10-2-99-157.ec2.internal:7 INFO hook.py:486] Hook is writing from the hook with pid: 7
[1]#011train-rmse:3.52571#011validation-rmse:3.53473
[2]#011train-rmse:3.18357#011validation-rmse:3.19135
[3]#011train-rmse:2.87753#011validation-rmse:2.88396
[4]#011train-rmse:2.60175#011validation-rmse:2.60792
[5]#011train-rmse:2.35539#011validation-rmse:2.36057
[6]#011train-rmse:2.13328#011validation-rmse:2.13766
[7]#011train-rmse:1.93436#011validation-rmse:1.93815
[8]#011train-rmse:1.75541#011validation-rmse:1.75955
[9]#011train-rmse:1.59684#011validation-rmse:1.60086
[10]#011train-rmse:1.45230#011validation-rmse:1.45613
[11]#011train-rmse:1.32451#011validation-rmse:1.32875
```

```
[12]#011train-rmse:1.21094#011validation-rmse:1.21528
[13]#011train-rmse:1.10819#011validation-rmse:1.11280
[14]#011train-rmse:1.01308#011validation-rmse:1.01895
[15]#011train-rmse:0.93073#011validation-rmse:0.93658
[16]#011train-rmse:0.85905#011validation-rmse:0.86507
[17]#011train-rmse:0.79601#011validation-rmse:0.80297
[18]#011train-rmse:0.73982#011validation-rmse:0.74767
[19]#011train-rmse:0.69114#011validation-rmse:0.69955
[20]#011train-rmse:0.64512#011validation-rmse:0.65448
[21]#011train-rmse:0.60350#011validation-rmse:0.61416
[22]#011train-rmse:0.56996#011validation-rmse:0.58149
[23]#011train-rmse:0.53829#011validation-rmse:0.55041
[24]#011train-rmse:0.50812#011validation-rmse:0.52163
[25]#011train-rmse:0.48574#011validation-rmse:0.50023
[26]#011train-rmse:0.46520#011validation-rmse:0.48054
[27]#011train-rmse:0.44799#011validation-rmse:0.46408
[28]#011train-rmse:0.43257#011validation-rmse:0.44928
[29]#011train-rmse:0.41522#011validation-rmse:0.43304
[30]#011train-rmse:0.40289#011validation-rmse:0.42101
[31]#011train-rmse:0.39077#011validation-rmse:0.40964
[32]#011train-rmse:0.37809#011validation-rmse:0.39733
[33]#011train-rmse:0.36953#011validation-rmse:0.38971
[34]#011train-rmse:0.36220#011validation-rmse:0.38277
[35]#011train-rmse:0.35689#011validation-rmse:0.37801
[36]#011train-rmse:0.35162#011validation-rmse:0.37343
[37]#011train-rmse:0.34744#011validation-rmse:0.36965
[38]#011train-rmse:0.33678#011validation-rmse:0.35955
[39]#011train-rmse:0.33362#011validation-rmse:0.35671
[40]#011train-rmse:0.32949#011validation-rmse:0.35275
[41]#011train-rmse:0.32688#011validation-rmse:0.35081
[42]#011train-rmse:0.32397#011validation-rmse:0.34836
[43]#011train-rmse:0.32150#011validation-rmse:0.34640
[44]#011train-rmse:0.31763#011validation-rmse:0.34293
[45]#011train-rmse:0.31592#011validation-rmse:0.34152
[46]#011train-rmse:0.31395#011validation-rmse:0.33995
[47]#011train-rmse:0.30691#011validation-rmse:0.33306
[48]#011train-rmse:0.30479#011validation-rmse:0.33161
[49]#011train-rmse:0.30333#011validation-rmse:0.33069
[50]#011train-rmse:0.30207#011validation-rmse:0.32993
[51]#011train-rmse:0.30066#011validation-rmse:0.32894
[52]#011train-rmse:0.29921#011validation-rmse:0.32799
[53]#011train-rmse:0.29513#011validation-rmse:0.32431
```

```
[54]#011train-rmse:0.29353#011validation-rmse:0.32319
[55]#011train-rmse:0.29248#011validation-rmse:0.32286
[56]#011train-rmse:0.28842#011validation-rmse:0.31930
[57]#011train-rmse:0.28736#011validation-rmse:0.31863
[58]#011train-rmse:0.28610#011validation-rmse:0.31764
[59]#011train-rmse:0.28490#011validation-rmse:0.31707
[60]#011train-rmse:0.28445#011validation-rmse:0.31689
[61]#011train-rmse:0.28325#011validation-rmse:0.31600
[62]#011train-rmse:0.28005#011validation-rmse:0.31322
[63]#011train-rmse:0.27714#011validation-rmse:0.31055
[64]#011train-rmse:0.27636#011validation-rmse:0.31019
[65]#011train-rmse:0.27531#011validation-rmse:0.30966
[66]#011train-rmse:0.27495#011validation-rmse:0.30961
[67]#011train-rmse:0.27261#011validation-rmse:0.30751
[68]#011train-rmse:0.27181#011validation-rmse:0.30719
[69]#011train-rmse:0.27086#011validation-rmse:0.30673
[70]#011train-rmse:0.26922#011validation-rmse:0.30527
[71]#011train-rmse:0.26858#011validation-rmse:0.30505
[72]#011train-rmse:0.26801#011validation-rmse:0.30484
[73]#011train-rmse:0.26709#011validation-rmse:0.30413
[74]#011train-rmse:0.26644#011validation-rmse:0.30378
[75]#011train-rmse:0.26581#011validation-rmse:0.30345
[76]#011train-rmse:0.26528#011validation-rmse:0.30328
[77]#011train-rmse:0.26492#011validation-rmse:0.30306
[78]#011train-rmse:0.26347#011validation-rmse:0.30183
[79]#011train-rmse:0.26262#011validation-rmse:0.30145
[80]#011train-rmse:0.26190#011validation-rmse:0.30091
[81]#011train-rmse:0.26118#011validation-rmse:0.30070
[82]#011train-rmse:0.25986#011validation-rmse:0.29942
[83]#011train-rmse:0.25921#011validation-rmse:0.29912
[84]#011train-rmse:0.25777#011validation-rmse:0.29817
[85]#011train-rmse:0.25732#011validation-rmse:0.29794
[86]#011train-rmse:0.25673#011validation-rmse:0.29748
[87]#011train-rmse:0.25604#011validation-rmse:0.29707
[88]#011train-rmse:0.25562#011validation-rmse:0.29716
[89]#011train-rmse:0.25403#011validation-rmse:0.29586
[90]#011train-rmse:0.25352#011validation-rmse:0.29580
[91]#011train-rmse:0.25343#011validation-rmse:0.29572
[92]#011train-rmse:0.25298#011validation-rmse:0.29547
[93]#011train-rmse:0.25208#011validation-rmse:0.29489
[94]#011train-rmse:0.25175#011validation-rmse:0.29476
[95]#011train-rmse:0.25083#011validation-rmse:0.29410
```

```
[96]#011train-rmse:0.25074#011validation-rmse:0.29411
[97]#011train-rmse:0.25040#011validation-rmse:0.29390
[98]#011train-rmse:0.24988#011validation-rmse:0.29370
[99]#011train-rmse:0.24930#011validation-rmse:0.29357
[100]#011train-rmse:0.24832#011validation-rmse:0.29292
[101]#011train-rmse:0.24797#011validation-rmse:0.29286
[102]#011train-rmse:0.24773#011validation-rmse:0.29273
[103]#011train-rmse:0.24704#011validation-rmse:0.29230
[104]#011train-rmse:0.24644#011validation-rmse:0.29190
[105]#011train-rmse:0.24616#011validation-rmse:0.29185
[106]#011train-rmse:0.24597#011validation-rmse:0.29174
[107]#011train-rmse:0.24552#011validation-rmse:0.29166
[108]#011train-rmse:0.24511#011validation-rmse:0.29155
[109]#011train-rmse:0.24450#011validation-rmse:0.29122
[110]#011train-rmse:0.24392#011validation-rmse:0.29101
[111]#011train-rmse:0.24379#011validation-rmse:0.29100
[112]#011train-rmse:0.24358#011validation-rmse:0.29091
[113]#011train-rmse:0.24339#011validation-rmse:0.29079
[114]#011train-rmse:0.24262#011validation-rmse:0.29032
[115]#011train-rmse:0.24232#011validation-rmse:0.29033
[116]#011train-rmse:0.24182#011validation-rmse:0.28995
[117]#011train-rmse:0.24141#011validation-rmse:0.28982
[118]#011train-rmse:0.24083#011validation-rmse:0.28954
[119]#011train-rmse:0.24025#011validation-rmse:0.28945
[120]#011train-rmse:0.23996#011validation-rmse:0.28938
[121]#011train-rmse:0.23951#011validation-rmse:0.28926
[122]#011train-rmse:0.23929#011validation-rmse:0.28921
[123]#011train-rmse:0.23915#011validation-rmse:0.28911
[124]#011train-rmse:0.23864#011validation-rmse:0.28879
[125]#011train-rmse:0.23817#011validation-rmse:0.28861
[126]#011train-rmse:0.23796#011validation-rmse:0.28842
[127]#011train-rmse:0.23744#011validation-rmse:0.28851
[128]#011train-rmse:0.23713#011validation-rmse:0.28845
[129]#011train-rmse:0.23696#011validation-rmse:0.28839
[130]#011train-rmse:0.23653#011validation-rmse:0.28820
[131]#011train-rmse:0.23607#011validation-rmse:0.28809
[132]#011train-rmse:0.23580#011validation-rmse:0.28810
[133]#011train-rmse:0.23528#011validation-rmse:0.28788
[134]#011train-rmse:0.23485#011validation-rmse:0.28795
[135]#011train-rmse:0.23462#011validation-rmse:0.28789
[136]#011train-rmse:0.23447#011validation-rmse:0.28780
[137]#011train-rmse:0.23430#011validation-rmse:0.28773
```

```
[138]#011train-rmse:0.23417#011validation-rmse:0.28769
[139]#011train-rmse:0.23404#011validation-rmse:0.28766
[140]#011train-rmse:0.23380#011validation-rmse:0.28770
[141]#011train-rmse:0.23335#011validation-rmse:0.28762
[142]#011train-rmse:0.23327#011validation-rmse:0.28762
[143]#011train-rmse:0.23316#011validation-rmse:0.28758
[144]#011train-rmse:0.23307#011validation-rmse:0.28758
[145]#011train-rmse:0.23256#011validation-rmse:0.28730
[146]#011train-rmse:0.23211#011validation-rmse:0.28700
[147]#011train-rmse:0.23156#011validation-rmse:0.28691
[148]#011train-rmse:0.23124#011validation-rmse:0.28682
[149]#011train-rmse:0.23111#011validation-rmse:0.28681

2023-07-27 21:49:11 Completed - Training job completed
Training seconds: 92
Billable seconds: 92
```

Deploy Model

In [19]:
```
# Ref: http://sagemaker.readthedocs.io/en/latest/estimators.html
predictor = estimator.deploy(initial_instance_count=1,
                             instance_type='ml.m5.xlarge',
                             endpoint_name = job_name)
```

```
INFO:sagemaker:Creating model with name: xgboost-bikerental-v1-2023-07-27-21-55-26-291
INFO:sagemaker:Creating endpoint-config with name xgboost-bikerental-v1
INFO:sagemaker:Creating endpoint with name xgboost-bikerental-v1
----!
```

Run Predictions

In [20]:
```
# SDK 2.0 serializers
from sagemaker.serializers import CSVSerializer
```

In [21]:
```
predictor.serializer = CSVSerializer()
```

In [26]:
```
predictor.predict([[3,0,1,2,28.7,33.335,79,12.998,2011,7,7,3]])
```

Out[26]:
```
b'3.773822784423828\n'
```

In [45]:
```
#DWB#  Another cell with the line below to save this result,
#DWB#+ so we can turn it into what we really want to predict
log_ish_pred = \
  float(predictor.predict(
            [[3,0,1,2,28.7,33.335,79,12.998,2011,7,7,3]])
        )

print(f"The prediction - as log(count_prediction + 1) - is: {log_ish_pred}")
type(log_ish_pred)
```

```
The prediction - as log(count_prediction + 1) - is: 3.773822784423828
```

Out[45]:
```
float
```

In [46]:
```
#DWB#  We're using the data with the log(x+1) instead of the count, x.
#DWB#+ So, to get the actual count predicted, we need exp(x) - 1
print(f"The prediction - as a count - is: {np.expm1(np.array(log_ish_pred))}")
```

```
The prediction - as a count - is: 42.546214847510505
```

Summary

1. Ensure Training, Test and Validation data are in S3 Bucket
2. Select Algorithm Container Registry Path - Path varies by region
3. Configure Estimator for training - Specify Algorithm container, instance count, instance type, model output location
4. Specify algorithm specific hyper parameters
5. Train model
6. Deploy model - Specify instance count, instance type and endpoint name
7. Run Predictions