

```

-----
Py4JJavaError                                Traceback (most recent call last)
<command-9101119> in <module>
      5  print('{} takes {} seconds'.format(phrase, round(end - start, 2)))
      6  return result
----> 7  model = with_benchmark('Training', lambda: classifier.fit(train_data))

<command-9101119> in with_benchmark(phrase, action)
      1  def with_benchmark(phrase, action):
      2  start = time()
----> 3  result = action()
      4  end = time()
      5  print('{} takes {} seconds'.format(phrase, round(end - start, 2)))

<command-9101119> in <lambda>()
      5  print('{} takes {} seconds'.format(phrase, round(end - start, 2)))
      6  return result
----> 7  model = with_benchmark('Training', lambda: classifier.fit(train_data))

/databricks/spark/python/pyspark/ml/base.py in fit(self, dataset, params)
    127  return self.copy(params)._fit(dataset)
    128  else:
--> 129  return self._fit(dataset)
    130  else:
    131  raise ValueError("Params must be either a param map or a list/tuple
of param maps, ")

/databricks/spark/python/pyspark/ml/wrapper.py in _fit(self, dataset)
    319
    320  def _fit(self, dataset):
--> 321  java_model = self._fit_java(dataset)
    322  model = self._create_model(java_model)
    323  return self._copyValues(model)

/databricks/spark/python/pyspark/ml/wrapper.py in _fit_java(self, dataset)
    316  """
    317  self._transfer_params_to_java()
--> 318  return self._java_obj.fit(dataset._jdf)
    319
    320  def _fit(self, dataset):

```

```

/databricks/spark/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py in
__call__(self, *args)
 1303     answer = self.gateway_client.send_command(command)
 1304     return_value = get_return_value(
-> 1305         answer, self.gateway_client, self.target_id, self.name)
 1306
 1307     for temp_arg in temp_args:

```

```

/databricks/spark/python/pyspark/sql/utils.py in deco(*a, **kw)
 126     def deco(*a, **kw):
 127         try:
--> 128             return f(*a, **kw)
 129     except py4j.protocol.Py4JJavaError as e:
 130         converted = convert_exception(e.java_exception)

```

```

/databricks/spark/python/lib/py4j-0.10.9-src.zip/py4j/protocol.py in
get_return_value(answer, gateway_client, target_id, name)
 326         raise Py4JJavaError(
 327             "An error occurred while calling {0}{1}{2}.\n".
--> 328             format(target_id, ".", name), value)
 329     else:
 330         raise Py4JError(

```

```

Py4JJavaError: An error occurred while calling o309.fit.
: ml.dmlc.xgboost4j.java.XGBoostError: XGBoostModel training failed
  at
ml.dmlc.xgboost4j.scala.spark.XGBoost$.postTrackerReturnProcessing(XGBoost.s
cala:848)
  at ml.dmlc.xgboost4j.scala.spark.XGBoost$.
$anonfun$trainDistributedPreferGpu$1(XGBoost.scala:656)
  at ml.dmlc.xgboost4j.scala.spark.XGBoost$.
$anonfun$trainDistributedPreferGpu$1$adapted(XGBoost.scala:636)
  at scala.collection.TraversableLike.
$anonfun$map$1(TraversableLike.scala:238)
  at scala.collection.immutable.List.foreach(List.scala:392)
  at scala.collection.TraversableLike.map(TraversableLike.scala:238)
  at scala.collection.TraversableLike.map$(TraversableLike.scala:231)
  at scala.collection.immutable.List.map(List.scala:298)
  at
ml.dmlc.xgboost4j.scala.spark.XGBoost$.trainDistributedPreferGpu(XGBoost.scal
a:635)
  at
ml.dmlc.xgboost4j.scala.spark.XGBoostClassifier.trainHonorGpu(XGBoostClassifie

```

```
r.scala:266)
  at
ml.dmlc.xgboost4j.scala.spark.XGBoostClassifier.trainGpu(XGBoostClassifier.scala
:203)
  at
ml.dmlc.xgboost4j.scala.spark.XGBoostClassifier.fit(XGBoostClassifier.scala:191)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
  at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImp
l.java:43)
  at java.lang.reflect.Method.invoke(Method.java:498)
  at py4j.reflection.MethodInvoker.invoke(MethodInvoker.java:244)
  at py4j.reflection.ReflectionEngine.invoke(ReflectionEngine.java:380)
  at py4j.Gateway.invoke(Gateway.java:295)
  at
py4j.commands.AbstractCommand.invokeMethod(AbstractCommand.java:132)
  at py4j.commands.CallCommand.execute(CallCommand.java:79)
  at py4j.GatewayConnection.run(GatewayConnection.java:251)
  at java.lang.Thread.run(Thread.java:748)
```