

記事

[Toshihiko Minamoto](#) · 2021年4月8日 6m read

## Jupyter Notebook + Apache Spark + InterSystems IRIS を起動させる方法

皆さん、こんにちは。今日は、Jupyter Notebook をインストールして、Apache Spark と InterSystems IRIS に接続したいと思います。

注 記: 以下にお見せる作業は Ubuntu 18.04 で Python 3.6.5 を使って実行しました。

### はじめに

Apache Zeppelin の代わりに認知度が高く、よく普及していて、主に Python ユーザーの間で人気というノートブックをお探しの方は、Jupyter notebookをおすすめします。 Jupyter notebook は、とてもパワフルで優れたデータサイエンスツールです。大きなコミュニティが存在し、使用できるソフトウェアや連携先たくさんあります。Jupyter Notebook では、ライブコード、数式、視覚的インターフェース、ナレーションテキストを含む文書を作成できます。機能としてデータリネーミングや変換、数値シミュレーション、統計モデリング、データの視覚化、機械学習などが含まれています。最も重要なこととして、問題に直面したときにその解決を手伝ってくれる大きなコミュニティが存在します。

### 要件の確認

何かうまく行かないことがあれば、一番の「考えられる問題と解決策」をご覧ください。

まずは、Java 8 がインストールされていることを確認してください ( `java -version` で "1.8.x" が返される)。次に、 [apache spark](#) をダウンロードし、解凍します。それから、ターミナルで以下のコマンドを実行します。

```
pip3 install jupyter
```

```
pip3 install toree
```

```
jupyter toree install --spark_home=/path_to_spark/spark-2.3.1-bin-hadoop2.7 --interpreters=PySpark --user
```

では、ターミナルを離れ、 `vim ~/.bashrc` を実行してください。一番に次のコードをペーストします (これは環境変数です)。

```
export JAVA_HOME=/usr/lib/jvm/installed java 8
export PATH="$PATH:$JAVA_HOME/bin"
export SPARK_HOME=/path to spark/spark-2.3.1-bin-hadoop2.7
export PATH="$PATH:$SPARK_HOME/bin"
export PYSPARK_DRIVER_PYTHON=jupyter
export PYSPARK_DRIVER_PYTHON_OPTS="notebook"
```

```
File Edit View Search Terminal Help
1 .bashrc +
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

#my variables

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH="$PATH:$JAVA_HOME/bin"
export SPARK_HOME=/home/guardian/Desktop/spark-2.3.1-bin-hadoop2.7
export PATH="$PATH:$SPARK_HOME/bin"
export PYSARK_DRIVER_PYTHON=jupyter
export PYSARK_DRIVER_PYTHON_OPTS="notebook"
~
~
NORMAL .bashrc + 96% 1
```

それから、source ~/.bashrc を実行します。

### 正常に動作することを確認する

ここでは、Jupyter Notebook を起動しましょう。ターミナルで、`pyspark` を実行します。

```
File Edit View Search Terminal Help
guardian@guardian:~$ pyspark
[I 16:07:42.424 NotebookApp] Serving notebooks from local directory: /home/guardian
[I 16:07:42.424 NotebookApp] The Jupyter Notebook is running at:
[I 16:07:42.424 NotebookApp] http://localhost:8888/?token=4d74c58c87b45d7cf2f55673bbe523a9054c7a574acf3254
[I 16:07:42.424 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:07:42.425 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time, to login with a token:
http://localhost:8888/?token=4d74c58c87b45d7cf2f55673bbe523a9054c7a574acf3254
[I 16:07:43.174 NotebookApp] Accepting one-time-token-authenticated connection from 127.0.0.1
```

返された URL をブラウザで開きます。次の画像のような画面が表示されると思います。

Name	Last Modified	File size
algo	3 дня назад	
Desktop	день назад	
Documents	22 дня назад	
Downloads	4 часа назад	
metastore_db	2 часа назад	
Music	22 дня назад	
Pictures	3 минуты назад	
Public	22 дня назад	
snap	14 дней назад	
spark-warehouse	22 дня назад	
Templates	22 дня назад	
Videos	22 дня назад	

new をリセットし、Python 3 を選択したら、次のコードをパラグラフにペーストします。

```
import sys
print(sys.version)
sc
```

このような出力が見られるはずです。

```
In [2]: import sys
print(sys.version)
sc

3.6.5 (default, Apr 1 2018, 05:46:30)
[GCC 7.3.0]

Out[2]: SparkContext

Spark UI
Version
v2.3.1
Master
local[*]
AppName
PySparkShell

In [ ]:
```

ターミナルで `ctrl-c` を実行して Jupyter を停止します。

注意: 独自の jar ファイルを追加する場合は、好きな jar ファイルを `$SPARK_HOME/jars` に移動します。

`intersystems-jdbc` と `intersystems-spark` を使いたいので (jpmml ライブラリも必要)、必要な jar ファイルを Spark にコピーします。ターミナルで次のコードを実行します。

```
sudo cp /path to intersystems iris/dev/java/lib/JDK18/intersystems-jdbc-3.0.0.jar /path to
spark/spark-2.3.1-bin-hadoop2.7/jars
```

```
sudo cp /path to intersystems iris/dev/java/lib/JDK18/intersystems-spark-1.0.0.jar /path to spark/spark-2.3.1-bin-hadoop2.7/jars
```

```
sudo cp /path to jpmml/jpmml-sparkml-executable-version.jar /path to spark/spark-2.3.1-bin-hadoop2.7/jars
```

問題がないことを確認してください。タミカでもう一度次のコードを実行します。

pyspark を実行し、(前回の [記事](#) でご紹介した)

```
from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.clustering import KMeans
from pyspark.ml import Pipeline
from pyspark.ml.feature import RFormula
from pyspark2pmml import PMMLBuilder

dataFrame=spark.read.format("com.intersystems.spark"). /
option("url", "IRIS://localhost:51773/NAMESPACE").option("user", "dev"). /
option("password", "123"). /
option("dbtable", "DataMining.IrisDataset").load() # load iris dataset

(trainingData, testData) = dataFrame.randomSplit([0.7, 0.3]) # split the data into two sets
assembler = VectorAssembler(inputCols = ["PetalLength", "PetalWidth", "SepalLength", "SepalWidth"],
outputCol="features") # add a new column with features

kmeans = KMeans().setK(3).setSeed(2000) # clustering algorithm that we use

pipeline = Pipeline(stages=[assembler, kmeans]) # First, passed data will run against assembler and after
will run against kmeans.
modelKMeans = pipeline.fit(trainingData) # pass training data

pmmlBuilder = PMMLBuilder(sc, dataFrame, modelKMeans)
pmmlBuilder.buildFile("KMeans.pmml") # create pmml model
```

出力は次のようになります。

```
In [2]: from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.clustering import KMeans
from pyspark.ml import Pipeline
from pyspark.ml.feature import RFormula
from pyspark2pmml import PMMLBuilder

dataFrame=spark.read.format("com.intersystems.spark").\
option("url", "IRIS://localhost:51773/NEWSAMPLE").option("user", "dev").\
option("password", "123").\
option("dbtable", "DataMining.IrisDataset").load() # load iris dataset

(trainingData, testData) = dataFrame.randomSplit([0.7, 0.3]) # split the data into two sets
assembler = VectorAssembler(inputCols = ["PetalLength", "PetalWidth", "SepalLength", "SepalWidth"], outputCol="feature

kmeans = KMeans().setK(3).setSeed(2000) # clustering algorithm that we use

pipeline = Pipeline(stages=[assembler, kmeans]) # First, passed data will run against assembler and after will run aga
modelKMeans = pipeline.fit(trainingData) # pass training data

pmmlBuilder = PMMLBuilder(sc, dataFrame, modelKMeans)
pmmlBuilder.buildFile("KMeans.pmml") # create pmml model

Out[2]: '/home/guardian/KMeans.pmml'
```

出力ファイルが jpmml kmeans model になっていますので、すべて完璧です！

## 考えられる問題と解決策

- 'jupyter'コマンドが見つからない

1. vim ~/.bashrc;
2. 一番に export PATH="\$PATH:~/local/bin" を追加します。
3. ターミナルで 'source ~/.bashrc' を実行します。
4. 問題解決しない場合は pip3 と jupyter を再インストールしてください。

- env: 'jupyter': このようなファイルまたはディレクトリはありません。

1. ~/.bashrc で、'export PYSPARK\_DRIVER\_PYTHON=/home/.../local/bin/jupyter' を設定します。

- TypeError: 'JavaPackage' オブジェクトは呼び出せません。

1. 必要な jar ファイルが /.../spark-2.3.1-bin-hadoop2.7/jars にあることを確認します。
2. Notebook を再起動します。

- Java ゲットウェイアクセスのポート番号をドライバに送る前に壊してしまう

1. Java バージョンは 8 を使用する (Java 6/7 で動作すると思いますが、確認はしていません)。
2. echo \$JAVA\_HOME を実行すれば、Java のバージョン 8 が返されるはずですが、そうでない場合は、~/.bashrc のパスを変更します。
3. ターミナルに sudo update-alternatives --config java をペーストし、Java の適切なバージョンを選択します。
4. ターミナルに sudo update-alternatives --config javac をペーストし、Java の適切なバージョンを選択します。

- PermissionError: [Errno 13] Permission denied: '/usr/local/share/jupyter'

1. ターミナルでコマンドの最後に --user を追加します。

- Jupyter コマンド'toree'の実行エラー: [Errno 2] このようなファイルまたはディレクトリはありません。

1. sudo なしでコマンドを実行します。

- PYSPARK\_SUBMIT\_ARGS のようなシステム変数や他の spark/pyspark 変数を使用した場合、または /.../spark-2.3.1-bin-hadoop2.7/conf/spark-env.sh の変更が原因で、特定のエラーが生じの場合

1. これらの変数を削除して spark-env.sh を確認します。

## リンク

- [Jupyter](#)
- [Apache Toree](#)
- [Apache Spark](#)
- [ML モデルを InterSystems IRIS に読み込む](#)
- [Iris Dataset の K 平均法](#)
- [Apache Spark, Apache Zeppelin, InterSystems IRIS を起動させる方法](#)

[#AI](#) [#API](#) [#Python](#) [#互換性](#) [#初心者](#) [#InterSystems IRIS](#)

ソースURL: <https://jp.community.intersystems.com/post/jupyter-notebook-apache-spark-intersystems-iris-%E3%82%92%E8%B5%B7%E5%8B%95%E3%81%95%E3%81%9B%E3%82%8B%E6%96%B9%E6%B3%95>