Skills Network Labs

Account

File Edit View Run Kernel Git Tabs Settings Help

jupyterlab@jupyterlab-kays1 | final_project.ipynb | input-hmp.ipynb | spark-csv-to-parquet.ipynb | spark-train-lr.ipynb

Code | Run as Pipeline | Python

Filter files by name

/ ... / component-library / train /

Name	Last Modified
install.log	3 minutes ago
jpmml-sparkml-executable-1.5....	3 minutes ago
nvflare.ipynb	37 minutes ago
README.md	37 minutes ago
spark-train-lr.ipynb	a minute ago
spark-train-lr.yaml	37 minutes ago
train-mobilenet_v2.ipynb	37 minutes ago
train-mobilenet_v2.yaml	37 minutes ago

Spark Train Logistic Regression

Train Logistic Regression classifier with Apache SparkML

[1]:
```
%%bash
export version=`python --version |awk '{print $2}' |awk -F"." '{print $1$2}'`

echo $version

if [ $version == '36' ] || [ $version == '37' ]; then
    echo 'Starting installation...'
    pip3 install pyspark==2.4.8 wget==3.2 pyspark2pmml==0.5.1 > install.log 2> install.log
    if [ $? == 0 ]; then
        echo 'Please <<RESTART YOUR KERNEL>> (Kernel->Restart Kernel and Clear All Outputs)'
    else
        echo 'Installation failed, please check log:'
        cat install.log
    fi
elif [ $version == '38' ] || [ $version == '39' ]; then
    pip3 install pyspark==3.1.2 wget==3.2 pyspark2pmml==0.5.1 > install.log 2> install.log
    if [ $? == 0 ]; then
        echo 'Please <<RESTART YOUR KERNEL>> (Kernel->Restart Kernel and Clear All Outputs)'
    else
        echo 'Installation failed, please check log:'
        cat install.log
    fi
else
    echo 'Currently only python 3.6, 3.7 , 3.8 and 3.9 are supported, in case you need a different version please open an issue at https://github.com/IBM/claimed/issues'
    exit -1
fi
```

```
37
Starting installation...
Please <<RESTART YOUR KERNEL>> (Kernel->Restart Kernel and Clear All Outputs)
```

[2]:
```
from pyspark import SparkContext, SparkConf, SQLContext
import os
from pyspark.ml.classification import LogisticRegression
from pyspark.ml import Pipeline
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark2pmml import PMMLBuilder
from pyspark.ml.feature import StringIndexer
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.feature import MinMaxScaler
import logging
import shutil
import site
import sys
import wget
import re
```

[3]:
```
if sys.version[0:3] == '3.9':
    url = ('https://github.com/jpmml/jpmml-sparkml/releases/download/1.7.2/'
           'jpmml-sparkml-executable-1.7.2.jar')
    wget.download(url)
    shutil.copy('jpmml-sparkml-executable-1.7.2.jar',
                site.getsitepackages()[0] + '/pyspark/jars/')
elif sys.version[0:3] == '3.8':
    url = ('https://github.com/jpmml/jpmml-sparkml/releases/download/1.7.2/'
           'jpmml-sparkml-executable-1.7.2.jar')
    wget.download(url)
    shutil.copy('jpmml-sparkml-executable-1.7.2.jar',
                site.getsitepackages()[0] + '/pyspark/jars/')
elif sys.version[0:3] == '3.7':
    url = ('https://github.com/jpmml/jpmml-sparkml/releases/download/1.5.12/'
           'jpmml-sparkml-executable-1.5.12.jar')
    wget.download(url)
elif sys.version[0:3] == '3.6':
    url = ('https://github.com/jpmml/jpmml-sparkml/releases/download/1.5.12/'
           'jpmml-sparkml-executable-1.5.12.jar')
    wget.download(url)
else:
    raise Exception('Currently only python 3.6 , 3.7, 3,8 and 3.9 is supported, in case '
                    'you need a different version please open an issue at '
                    'https://github.com/IBM/claimed/issues')
```

[4]:
```
data_parquet = os.environ.get('data_parquet',
                              'data.parquet')  # input file name (parquet)
master = os.environ.get('master',
                        "local[*]")  # URL to Spark master
model_target = os.environ.get('model_target',
                              "model.xml")  # model output file name
data_dir = os.environ.get('data_dir',
                          '../../data/')  # temporary directory for data
input_columns = os.environ.get('input_columns',
                               '["x", "y", "z"]')  # input columns to consider
```

[5]:
```
parameters = list(
    map(lambda s: re.sub('$', '"', s),
        map(
            lambda s: s.replace('=', '="'),
            filter(
                lambda s: s.find('=') > -1 and bool(re.match(r'[A-Za-z0-9_]*=[.\/A-Za-z0-9]*', s)),
                sys.argv
            )
    )))

for parameter in parameters:
    logging.warning('Parameter: ' + parameter)
    exec(parameter)
```

[6]:
```
conf = SparkConf().setMaster(master)
#if sys.version[0:3] == '3.6' or sys.version[0:3] == '3.7':
conf.set("spark.jars", 'jpmml-sparkml-executable-1.5.12.jar')

sc = SparkContext.getOrCreate(conf)
sqlContext = SQLContext(sc)
spark = sqlContext.sparkSession
```

```
23/03/25 14:29:36 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/03/25 14:29:40 WARN util.Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
23/03/25 14:29:40 WARN util.Utils: Service 'SparkUI' could not bind on port 4041. Attempting port 4042.
```

[7]:
```
df = spark.read.parquet(data_dir + data_parquet)
```

[8]:
```
# register a corresponding query table
df.createOrReplaceTempView('df')
```

[9]:
```
from pyspark.sql.types import DoubleType
df = df.withColumn("x", df.x.cast(DoubleType()))
df = df.withColumn("y", df.y.cast(DoubleType()))
df = df.withColumn("z", df.z.cast(DoubleType()))
```

[10]:
```
splits = df.randomSplit([0.8, 0.2])
df_train = splits[0]
df_test = splits[1]
```

[11]:
```
indexer = StringIndexer(inputCol="class", outputCol="label")

vectorAssembler = VectorAssembler(inputCols=eval(input_columns),
```

Support/Feedback

```
normalizer = MinMaxScaler(inputCol="features", outputCol="features_norm")
```

[12]:
```
lr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)
```

[13]:
```
pipeline = Pipeline(stages=[indexer, vectorAssembler, normalizer, lr])
```

[14]:
```
model = pipeline.fit(df_train)
```

```
23/03/25 14:30:41 WARN netlib.BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
23/03/25 14:30:41 WARN netlib.BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS
```

[15]:
```
prediction = model.transform(df_train)
```

[16]:
```
binEval = MulticlassClassificationEvaluator(). \
    setMetricName("accuracy"). \
    setPredictionCol("prediction"). \
    setLabelCol("label")

binEval.evaluate(prediction)
```

[16]:
```
0.20738740606084372
```

[17]:
```
pmmlBuilder = PMMLBuilder(sc, df_train, model)
pmmlBuilder.buildFile(data_dir + model_target)
```

[17]:
```
'/resources/labs/BD0231EN/claimed/component-library/train/../../data/model.xml'
```

[ ]: