**课后作业-Spark入门实践**

学号：2017013207 姓名：丁泽锋 班级：电商1701

本次上机实践的内容如下：

1）Spark的安装与配置（Local模式或伪分布模式）；

2）Spark-Shell的使用。尝试在Spark-Shell上以交互式编程的模式完成WordCount（输入可以是本地文件，如“file:///usr/local/spark/README.md”），要求按照字符串出现的次数倒序输出；

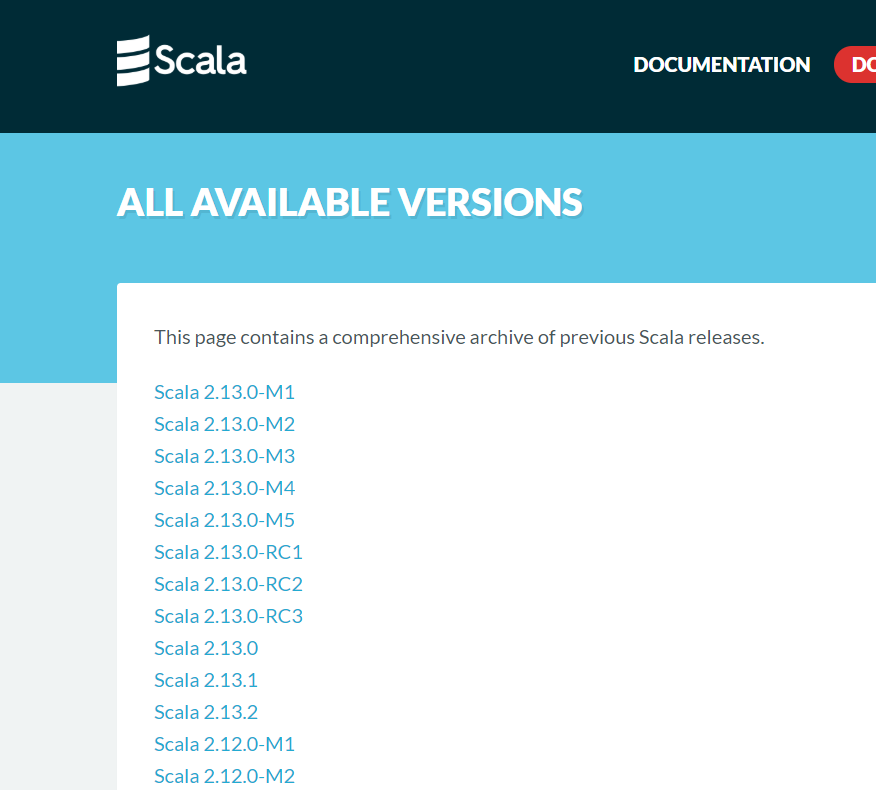
3）Spark API的使用。基于Spark API编写程序，语言可以是Java、Scala或Python。（此项作为挑战）

注：切忌照抄。有困难可告知。

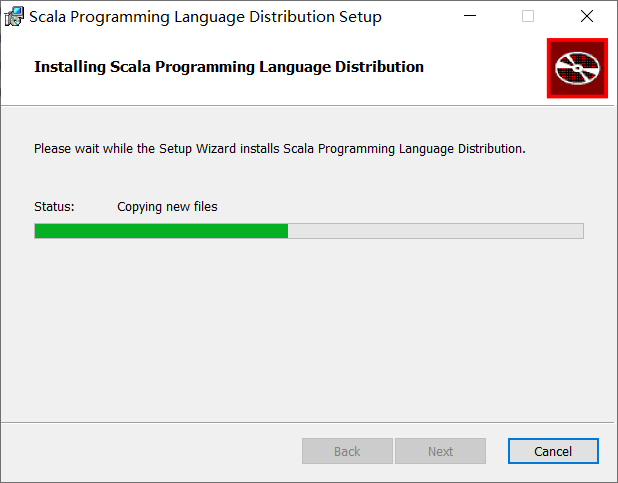
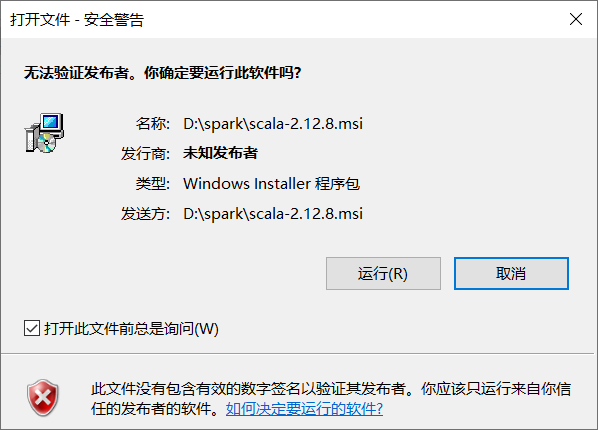
# 一、Spark的安装与配置

## 1. Scala的安装

从官网下载安装文件，官网网址为：<https://www.scala-lang.org/download/all.html>



运行安装文件，完成对Scala的安装。

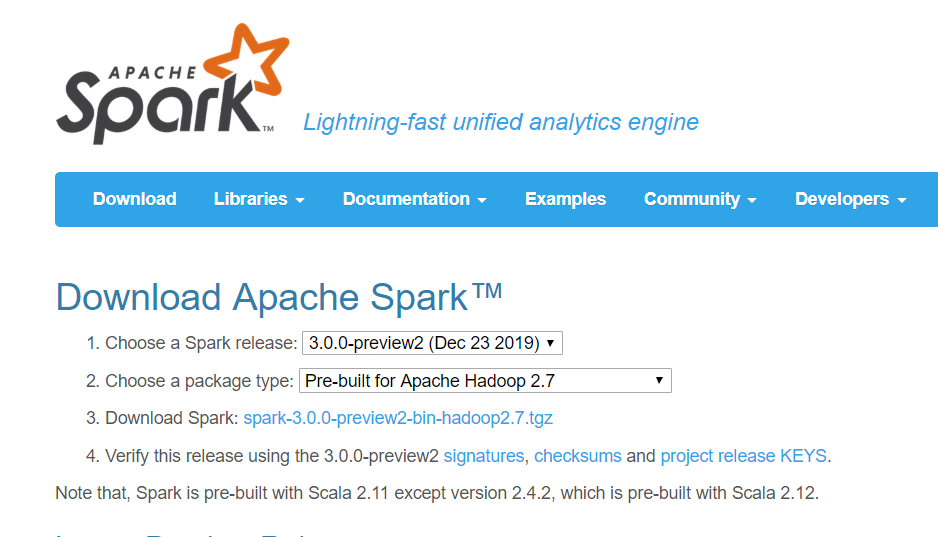


Scala安装好之后，会自动在环境变量PATH里面配置。打开控制台，输入Scala，验证是否安装完成。

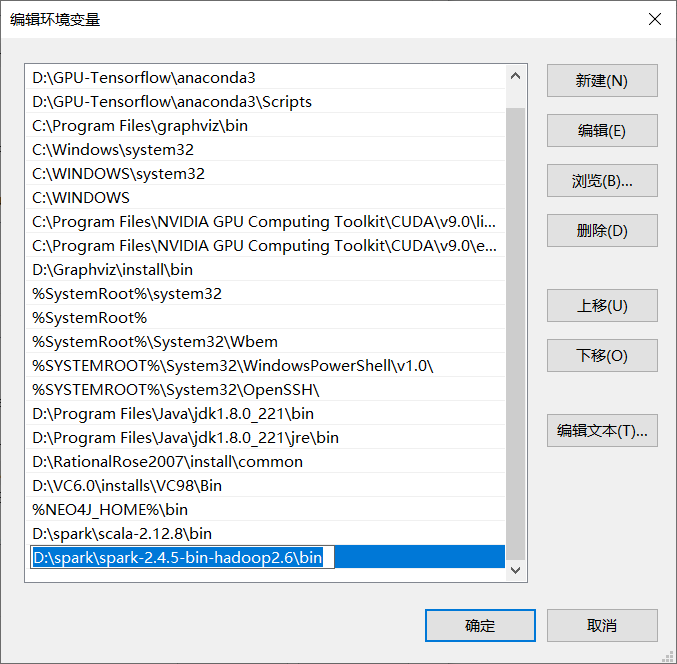


## 2. Spark的安装

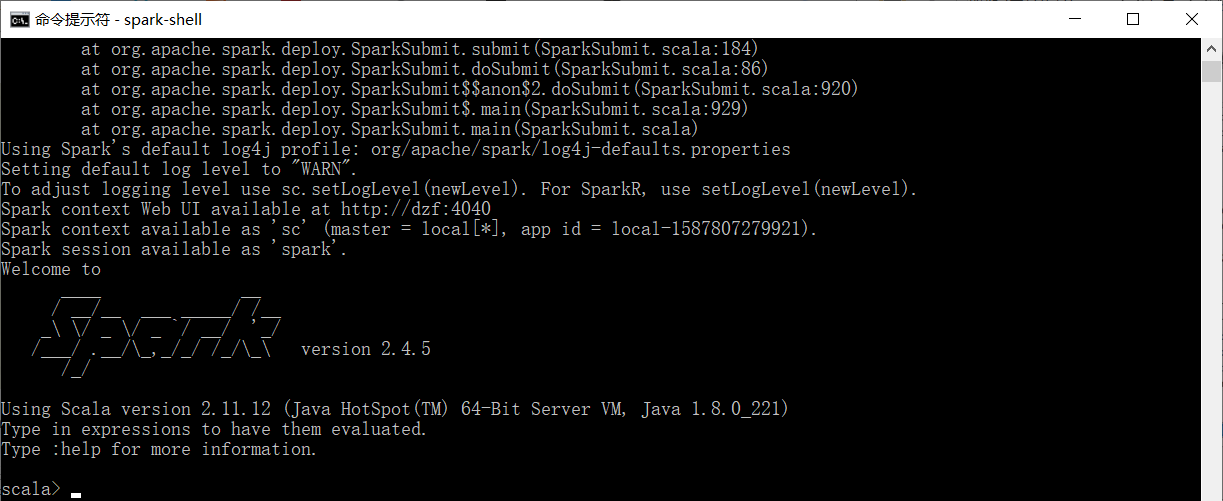
从Spark官网下载Spark安装包，官网网址为：<http://spark.apache.org/downloads.html>



解压下载好的文件，并配置环境变量。

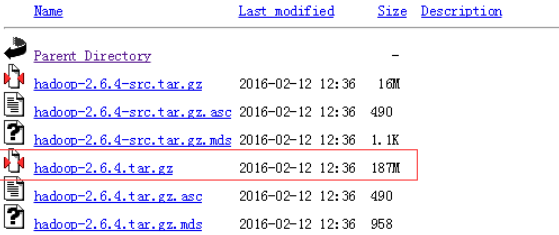


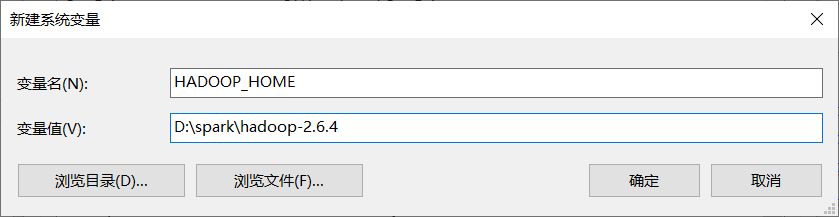
验证spark是否能正常启动。在控制台输入命令：spark-shell



## 2. Hadoop的安装

下载Hadoop-2.6.4压缩包，并减压

  
设置环境变量：



Path变量：

进行测试：

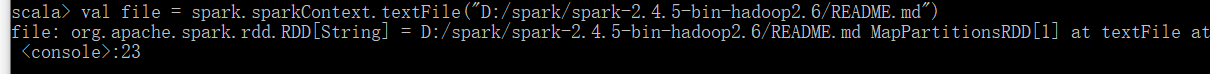


Spark开发环境安装完成。

# 二、在Spark-Shell上完成WordCount

读入文件，代码为：

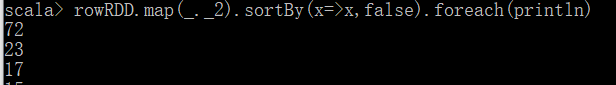
val file = spark.sparkContext.textFile("D:/spark/spark-2.4.5-bin-hadoop2.6/README.md")



val rowRDD = file.flatMap(\_.split(" ")).map(x=>(x,1)).reduceByKey((x,y)=>(x+y))

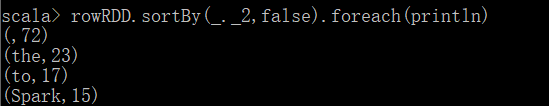


rowRDD.map(\_.\_2).sortBy(x=>x,false).foreach(println)



进行倒序输出：

rowRDD.sortBy(\_.\_2,false).foreach(println)



# Spark API的使用

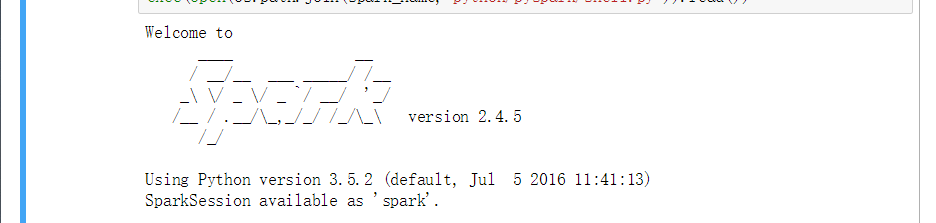
结合科创时学习Python及房价预测项目的实践经历，采用PySpark来进行Spark API的实践。实践方式通过一个Demo（波士顿房价预测模型）来展示，通过分析房子的不同信息（面积，区位等）预测房子的价格。

采用（Jupyter Notebook +PySpark+其余python开发的必要的包）开发实践该小项目。

代码块1：



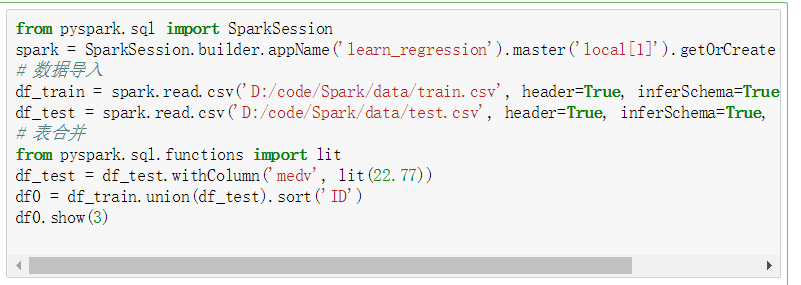
执行结果：



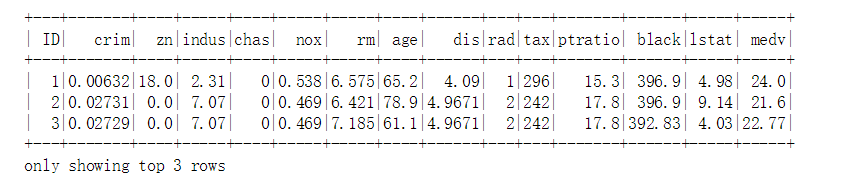
注释：

导入PySpark框架。

代码块2：



执行结果：



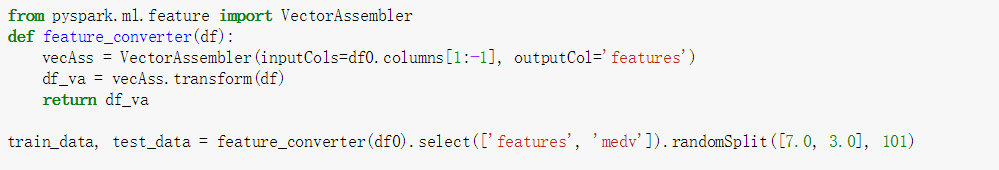
注释：

查看数据集信息：

|  |  |
| --- | --- |
| CRIM | 城镇人均犯罪率 |
| ZN | 占地面积超过25,000平方英尺的住宅用地比例 |
| INDUS | 每个城镇非零售业务的比例 |
| CHAS | Charles River虚拟变量（如果河流经过则= 1;否则为0） |
| NOX | 氮氧化物浓度（每千万份） |
| RM | 每间住宅的平均房间数 |
| AGE | 1940年以前建造的自住单位比例 |
| DIS | 加权平均值到五个波士顿就业中心的距离 |
| RAD | 径向高速公路的可达性指数 |
| TAX | 每10,000美元的全额物业税率 |
| PTRATIO | 城镇的学生与教师比例 |
| BLACK | 1000（Bk - 0.63）²其中Bk是城镇黑人的比例 |
| LSTAT | 人口较低的地位（百分比） |
| MEDV | 自住房屋的中位数价值1000美元。这是目标变量。 |

数据集包括两个部分，train 和 test ，test中没有相关medv的数据，都用22.77代替后，将train 和 test中数据合并，方便评估模型。

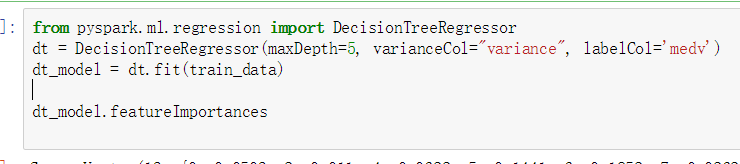
代码块3：



注释：

使用 feature 模块的 VectorAssembler 将特征合并，然后按7:3分为训练集和测试集：

代码块4：



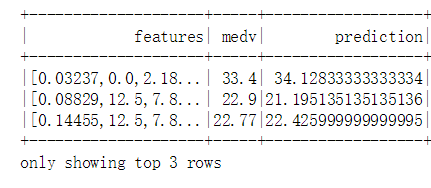
注释：

使用决策树回归模型进行训练。

代码块5：



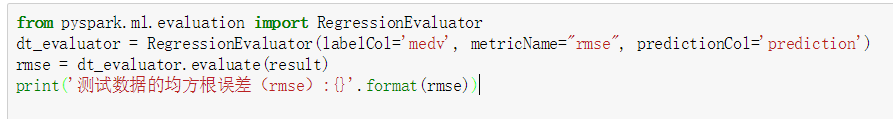
执行结果：



注释：

展示三条模型的预测结果，medv为真实值，prediction为预测值。

代码块6：



执行结果：



注释：

显示模型的均方根误差（表示真实值与其模拟值之间的偏差），用于模型评估。

备注：