|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实践项目- 基于指定数据集的统计与分析项目介绍Spark与Python结合 Python是数据分析最常用的语言之一，而Apache Spark是一个开源的强大的分布式查询和处理引擎。本实验讲解如何基于Python语言进行Spark Application编程，完成数据获取、处理、数据分析及可视化方面常用的数据分析方法与技巧，通过实验操作让学生掌握使用PySpark来分析数据。  通过初始安装Hadoop、Spark集群，并考虑到了当前应用场景需要支持Python程序运行在Spark集群之上，这时可以准备好对应Python软件包、依赖模块，在Hadoop集群中的每个节点上进行安装。这样，Hadoop集群的每个NodeManager上都具有Python环境，可以编写PySpark Application并在集群上运行。目前比较流行的是直接安装Python虚拟环境，使用Anaconda等软件，可以极大地简化Python环境的管理工作。  这种方式的缺点是，如果后续使用Python编写Spark Application，需要增加新的依赖模块，那么就需要在Hadoop集群的每个节点上都进行该新增模块的安装。而且，如果依赖Python的版本，可能还需要管理不同版本Python环境。目前实验Hadoop使用单机模式安装，如果数据量过大，学生电脑配置不高，会发生卡死现象。 实验设计 针对DataExpo2009数据集，通过Python、PySpark等工具，进行航空公司延误和取消分析。  该数据集包含1987年10月到2008年12月美国境内所有商业航班的航班到达和离开详细信息。这是一个大型数据集：总共有近1.2亿条记录，占用了1.6 GB的压缩空间和12 GB的未压缩时空间。  针对数据量太大的情况，可以选择kaggle上另一数据集DelayedFlights.csv进行分析，此数据集有1936758条数据，共有19家航空公司，是截取DataExpo2009一部分数据组成，有相同的数据字典。可选择前10个数据量最多的航空公司：   |  |  | | --- | --- | | **航空公司** | **数据量** | | WN | 377602 | | AA | 191865 | | MQ | 141920 | | UA | 141426 | | OO | 132433 | | DL | 114238 | | XE | 103663 | | CO | 100195 | | US | 98425 | | EV | 81877 |   可参考选择。  航班有5205种类，最多的航班前10数据为：   |  |  | | --- | --- | | **航班** | **数据量** | | LAX-SFO | 4739 | | ORD-LGA | 4396 | | ATL-LGA | 4058 | | SFO-LAX | 4020 | | LAS-LAX | 3516 | | LGA-ATL | 3354 | | ORD-EWR | 3347 | | LGA-ORD | 3327 | | ATL-EWR | 3283 | | DFW-ORD | 3278 |   可参考选择。  若是觉得193万条数据较少，也可选择DataExpo2009数据集中2008年的数据集，有7009728条数据。  实验需要读入并处理数据（可以存入HDFS、HBase等），然后制作一些图形和表格来进行数据分析，能更清楚地了解这个数据集中的内容。比如进行以下分析：  (1) 查看飞机延误时间最长的前10名航班。  (2) 计算延误的和没有延误的航空公司的比例。  (3) 分析一天中、一周中延误最严重的飞行时间。  (4) 短途航班和长途航班，哪种航班取消更严重？  (5) 建立机器学习算法模型，预测未来航班取消情况。 模型选择 模型可选择：支持向量机（SVM）、随机森林（Random Forest）、梯度提升树(GBDT)、线性判别分析（Linear Discriminant Analysis）、伯努利贝叶斯分类（BernoulliNB）、Adaboost、XGBoost等。  模型评估方法可用：准确率和召回率、ROC 曲线、AUC（ROC 曲线下的面积）等，按照评分高低进行比较。  当然，如果要用深度学习来训练算法模型，由于数据集涉及到时间序列相关，建议选择长短期记忆模型（LSTM）。 DataExpo2009数据字典  |  |  |  | | --- | --- | --- | |  | **Name** | **Description** | | **1** | **Year** | **1987-2008** | | **2** | **Month** | **1-12** | | **3** | **DayofMonth** | **1-31** | | **4** | **DayOfWeek** | **1 (Monday) - 7 (Sunday)** | | **5** | **DepTime** | **actual departure time (local, hhmm)** | | **6** | **CRSDepTime** | **scheduled departure time (local, hhmm)** | | **7** | **ArrTime** | **actual arrival time (local, hhmm)** | | **8** | **CRSArrTime** | **scheduled arrival time (local, hhmm)** | | **9** | **UniqueCarrier** | **unique carrier code** | | **10** | **FlightNum** | **flight number** | | **11** | **TailNum** | **plane tail number** | | **12** | **ActualElapsedTime** | **in minutes** | | **13** | **CRSElapsedTime** | **in minutes** | | **14** | **AirTime** | **in minutes** | | **15** | **ArrDelay** | **arrival delay, in minutes** | | **16** | **DepDelay** | **departure delay, in minutes** | | **17** | **Origin** | **origin IATA airport code** | | **18** | **Dest** | **destination IATA airport code** | | **19** | **Distance** | **in miles** | | **20** | **TaxiIn** | **taxi in time, in minutes** | | **21** | **TaxiOut** | **taxi out time in minutes** | | **22** | **Cancelled** | **was the flight cancelled?** | | **23** | **CancellationCode** | **reason for cancellation (A = carrier, B = weather, C = NAS, D = security)** | | **24** | **Diverted** | **1 = yes, 0 = no** | | **25** | **CarrierDelay** | **in minutes** | | **26** | **WeatherDelay** | **in minutes** | | **27** | **NASDelay** | **in minutes** | | **28** | **SecurityDelay** | **in minutes** | | **29** | **LateAircraftDelay** | **in minutes** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 环境要求（另附录有完全基于jupyter notebook的环境配置）1. Java （需要安装1.6.x及其以上版本)2. Hadoop和Spark（前两次实验已有）3. PySpark、Jupyter或者Pycharm 在这里，我们使用了Hadoop V3.1.4版本以及Spark-3.0.1版本。  3.1 PySpark的安装及测试  首先确认Spark的环境是否配置完成（实验一要求的spark环境）   |  | | --- | | $ su - hadoop  口令输入：hadoop  $ sudo gedit ~/.bash\_profile  看到配置文件里有如下spark环境内容    $ echo $SPARK \_HOME  /hadoop/app/spark  上述输出确认hadoop的环境变量设置有效，如果无效则激活环境变量：  $ source ~/.bash\_profile |   测试PySpark是否可以成功启动    PySpark有几种运行模式：   1. 可以在本地直接运行 2. 在Hadoop YARN上运行PySpark 3. 构建Spark Standalone集群，在Spark Standalone上运行PySpark.   3.2 Jupyter Notebook安装及测试  Jupyter Notebook具备交互式界面，我们可以在Web界面输入Python命令后立刻看到结果。我们还可将数据分析的过程和运行后的命令与结果存储成笔记本，下次可以打开笔记本，重新执行这些命令，IPython Notebook笔记本可以包含文字、数学公式、程序代码、结果、图形、视频。  通过安装Anaconda软件包直接安装Jupyter Notebook。   1. 下载Anaconda   https://repo.continuum.io/archive/index.html 进入网站后可以看到适用于Linux的安装包。可以选择Anaconda2-2.5.0-linux-x86\_64.sh，这个版本和Spark2.0兼容，或者以上的其他版本。    复制需要下载的anaconda的版本。  在终端使用wget命令   |  | | --- | | $ wget https://repo.anaconda.com/archive/Anaconda2-2.5.0-Linux-x86\_64.sh |   下载完成后，进行安装   |  | | --- | | $ bash Anaconda2-2.5.0-Linux-x86\_64.sh -b |   -b是批次安装，会忽略Licese条款。  加入PySpark及Anaconda路径   |  | | --- | | $sudo gedit ~/.bash\_profile |   在打开的文档中，加入以下设置：  export PATH=/hadoop/anaconda2/bin:$PATH  export ANACONDA\_PATH=/hadoop/anaconda2  export PYSPARK\_DRIVER\_PYTHON=$ANACONDA\_PATH/bin/ipython  export PYSPARK\_PYTHON=$ANACONDA\_PATH/bin/python  再次source   |  | | --- | | $ source ~/.bash\_profile |  1. 尝试在Notebook中使用Spark   可以在“终端”程序中输入下列命令，创建并切换ipynotebook 工作目录。  $ Mkdir -p ~/pythonwork/ipynotebook  $ cd ~/pythonwork/ipynotebook  进入工作目录后，输入以下命令  $ PYSPARK\_DRIVER\_PYTHON=ipython PYSPARK\_DRIVER\_PYTHON\_OPTS="notebook" pyspark  然后从浏览器启动Notebook界面。  默认是以spark的本地模式运行。  例如在新建的notebook中，输入代码  sc.master  运行结果如下图所示 |

|  |
| --- |
| 实践内容要求 针对DataExpo2009数据集，通过Python、PySpark等工具，进行数据读取、数据处理及可视化展示，并进行以下分析：  (1) 查看飞机延误时间最长的前10名航班。  (2) 计算延误的和没有延误的航空公司的比例。  (3) 分析一天中、一周中延误最严重的飞行时间。  (4) 短途航班和长途航班，哪种航班取消更严重？  (5) 建立机器学习算法模型，预测未来航班取消情况。 |

|  |
| --- |
| 实践结果 （1）-（4）为统计分析展示结果，（5）为机器学习模型结果，请将结果代码和分析过程放在实验报告中 |

# 附录 PySpark安装配置

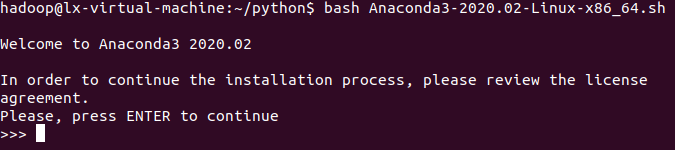
Spark安装好后，需要先安装python环境来配置pyspark。

下载anaconda包，这里选用的版本是Anaconda3-2020.02-Linux-x86\_64，下载地址：

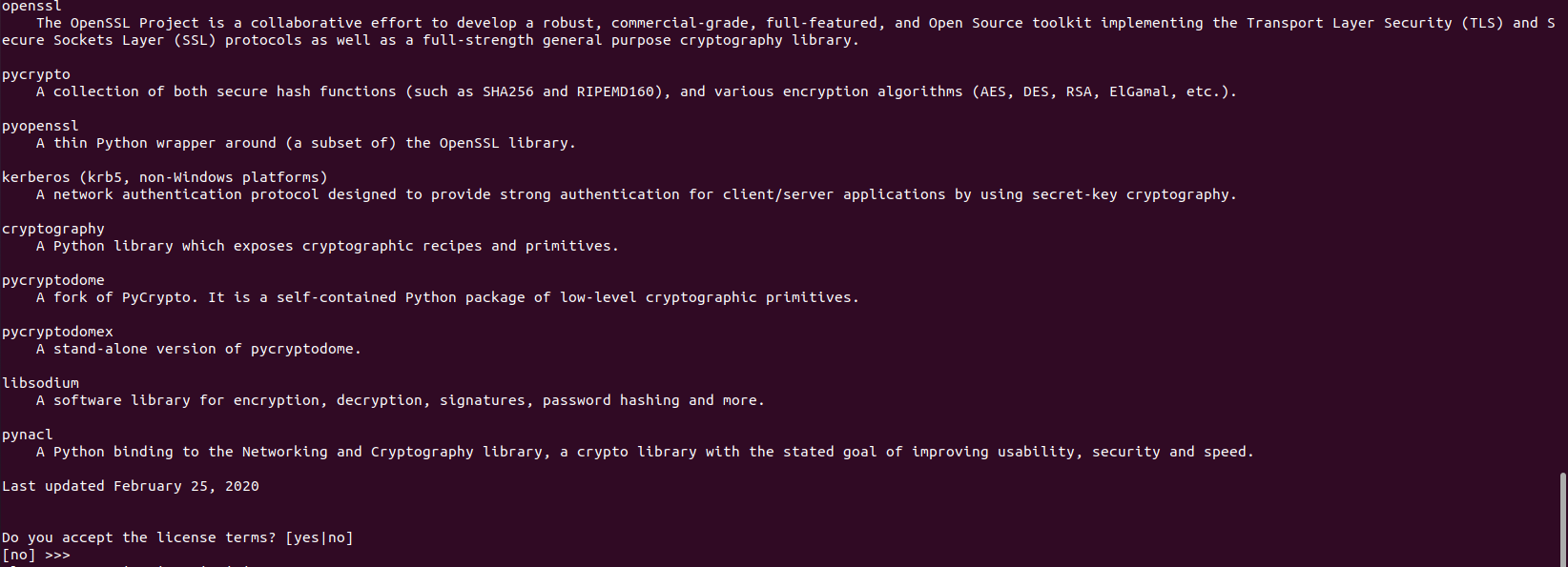
$ wget https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86\_64.sh

进入到安装包目录，执行命令：

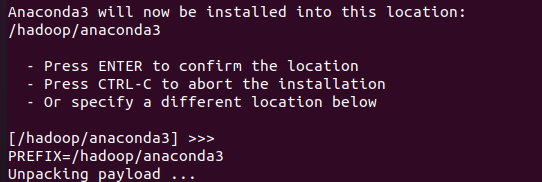
$ bash Anaconda3-2020.02-Linux-x86\_64



下一步输入回车键，并输入yes



下一步设置安装路径



安装好后设置环境变量：

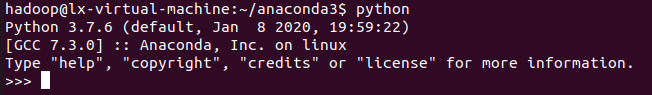
$ vi /hadoop/.bash\_profile

$ export PATH=$PATH:$SPARK\_HOME/bin:/hadoop/anaconda3/bin

保存关闭后，执行以下命令使得环境变量生效：

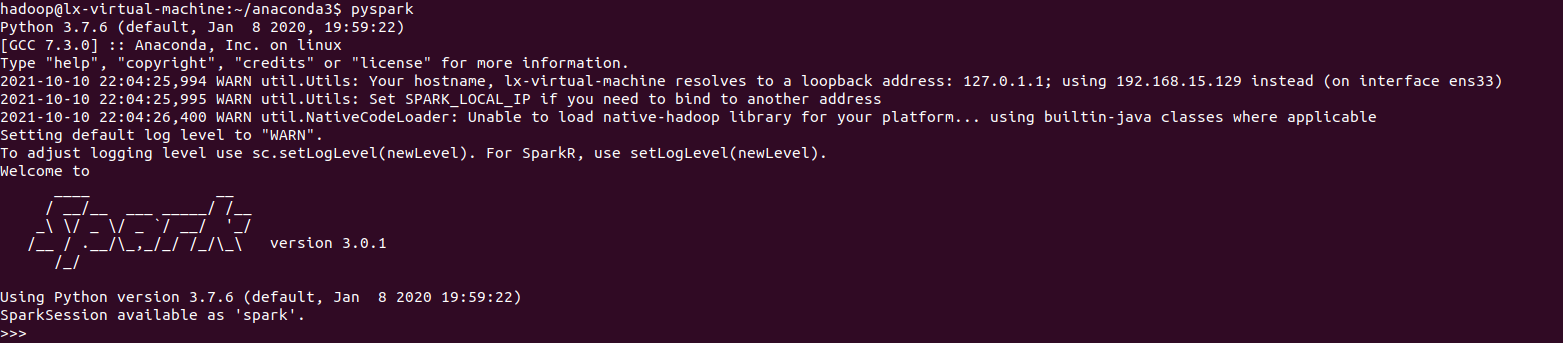
$ source /hadoop/.bash\_profile

输入python：



Python环境安装成功。

输入pyspark：



Pyspark环境配置成功。

## 二、Jupyter安装配置

Jupyter Notebook是基于网页的用于交互计算的应用程序。其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。简而言之，Jupyter Notebook是以网页的形式打开，可以在网页页面中直接编写代码和运行代码，代码的运行结果也会直接在代码块下显示。

配置PySpark driver

$ vi /hadoop/.bash\_profile

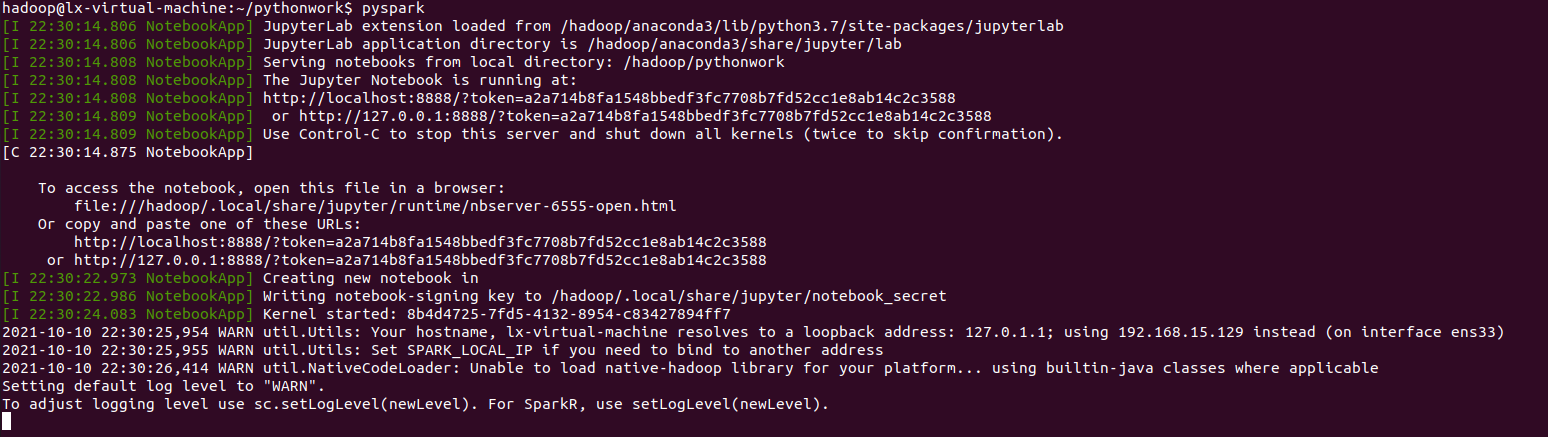
$ export PYSPARK\_DRIVER\_PYTHON=jupyter

$ export PYSPARK\_DRIVER\_PYTHON\_OPTS='notebook'

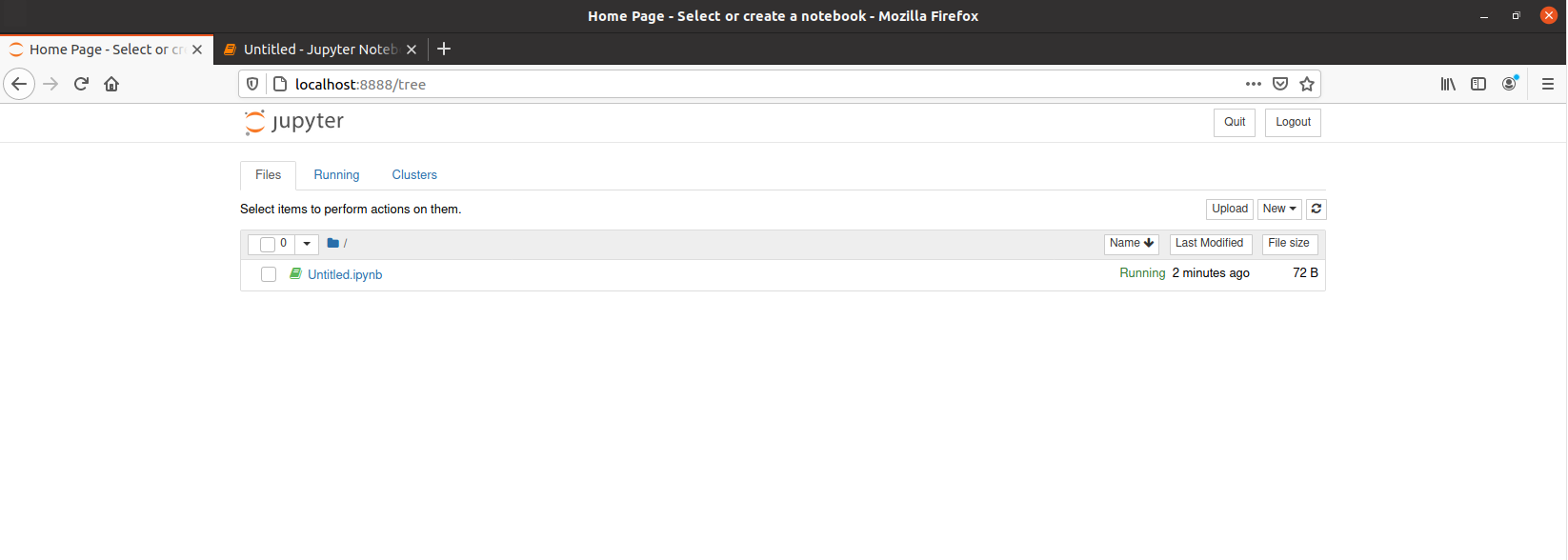
保存关闭后，执行以下命令使得环境变量生效：

$ source /hadoop/.bash\_profile

重新输入pyspark：



会自动启动jupyter：



创建python环境，就可以进行pyspark代码编写和运行了：

