大数据处理实验手册

概览

本次试验内容为：

本手册在 ubuntu18.04 平台上验证，相关软件环境为：

- JAVA 8
- Spark 3.0.2
- Scala 2.13.5

实验环境搭建

系统安装

下载 ubuntu18.04 镜像（你当然也可以使用别的 linux 版本）：

http://mirrors.aliyun.com/ubuntu-releases/18.04/ubuntu-18.04.5-desktop-amd64.iso

按正常流程安装虚拟机即可。（By the way: 有一个工具叫 virt-manager，个人觉得比较棒）

JAVA 环境配置

```
sudo apt-get install openjdk-8-jdk
java -version
```

```
penguin@penguin-Standard-PC-Q35-ICH9-2009:~$ java -version
openjdk version "1.8.0_282"
OpenJDK Runtime Environment (build 1.8.0_282-8u282-b08-0ubuntu1~18.04-b08)
OpenJDK 64-Bit Server VM (build 25.282-b08, mixed mode)
```

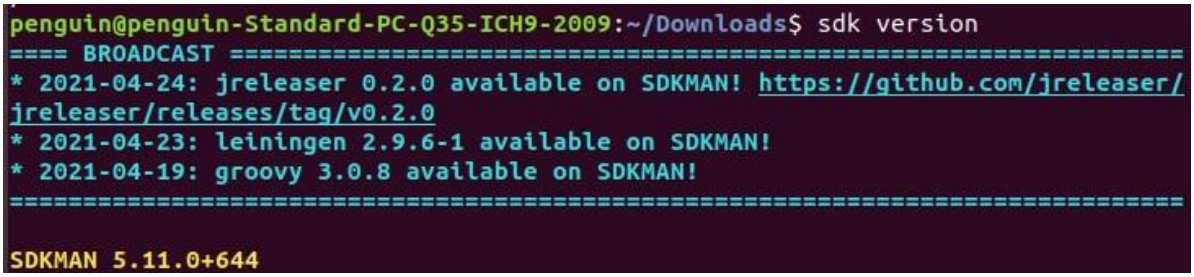
出现此提示信息表示安装完成。

scala 配置

> 本环节使用了第三方工具 SDKMAN 来帮助配置 scala，避免繁琐的环境变量配置。
>
> 如果你不能顺利完成这一步的操作，可以自行搜索一些 scala 安装教程

首先安装 SDKMAN

```
sudo apt install curl
curl -s "https://get.sdkman.io" | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
sdk version
```



然后使用 SDKMAN 来安装 scala

```
sdk install scala
```

Spark & hadoop 下载

下载 spark hadoop 合体安装包、解压：

https://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-3.0.2/spark-3.0.2-bin-hadoop3.2.tgz

```
tar -zxvf spark-3.0.2-bin-hadoop3.2.tgz
```

python 环境

```
sudo apt install python python-pip
pip install numpy
```

实验部分

本实验使用 python 调用 spark 进行 高斯混合模型（GMM）实验 – 车辆满意度预测

GMM 是一种概率式的聚类方法，属于生成式模型。对于给定的 K 个类别，采用 K 个高斯模型来表征各项数据的特征，计算各项数据归为各个类别的概率，生成 K 个聚类。通常被用于图像分类、音频识别等。

文件准备

在合适的地方（如你的 home 中）建立文件夹存放数据与代码

```
mkdir experiment
```

将 Car.txt 与 GMM_Spark.py 存入该文件夹中

Car.txt 描述（在代码中通过字典将字符串类型的值转换成 0，1，2，3 的整型值）

属性	值（离散）
车辆价格	{vhigh,high,med,low}
维修价格	{vhigh,high,med,low}
车门数量	{2,3,4,5more}
载客数	{2,4,more}
后备箱大小	{small,med,big}
安全性	{low,med,high}
满意度	{unacc,acc,vgood,good}

路径修正

编辑 GMM_Spark.py 文件，根据你之前创建的文件夹 experiment 的路径，将其中所有路径进行修改，例如。

```
df = sc.textFile("file:///usr/local/experiment/car.txt")
```

运行

进入解压后的 spark 文件夹（根据各位的下载路径而定）：

```
cd ~/work/spark-3.0.2-bin-hadoop3.2/bin
ls
```

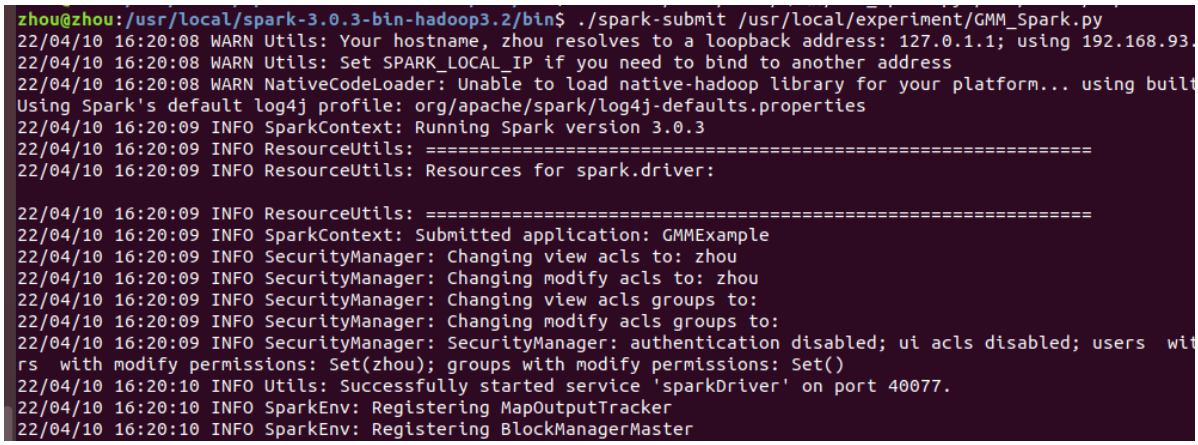
可以看到有很多工具。例如 pyspark 是与 python 的交互式编程环境、spark-shell 是与 scala 的交互式编程环境。可以自行尝试使用。

```
penguin@penguin-Standard-PC-Q35-ICH9-2009:~/work/spark-3.0.2-bin-hadoop3.2/bin$
ls
beeline                pyspark             spark-class.cmd    spark-sql
beeline.cmd            pyspark2.cmd        sparkR             spark-sql2.cmd
docker-image-tool.sh   pyspark.cmd         sparkR2.cmd        spark-sql.cmd
find-spark-home        run-example         sparkR.cmd         spark-submit
find-spark-home.cmd    run-example.cmd     spark-shell        spark-submit2.cmd
load-spark-env.cmd     spark-class         spark-shell2.cmd   spark-submit.cmd
load-spark-env.sh      spark-class2.cmd    spark-shell.cmd
```

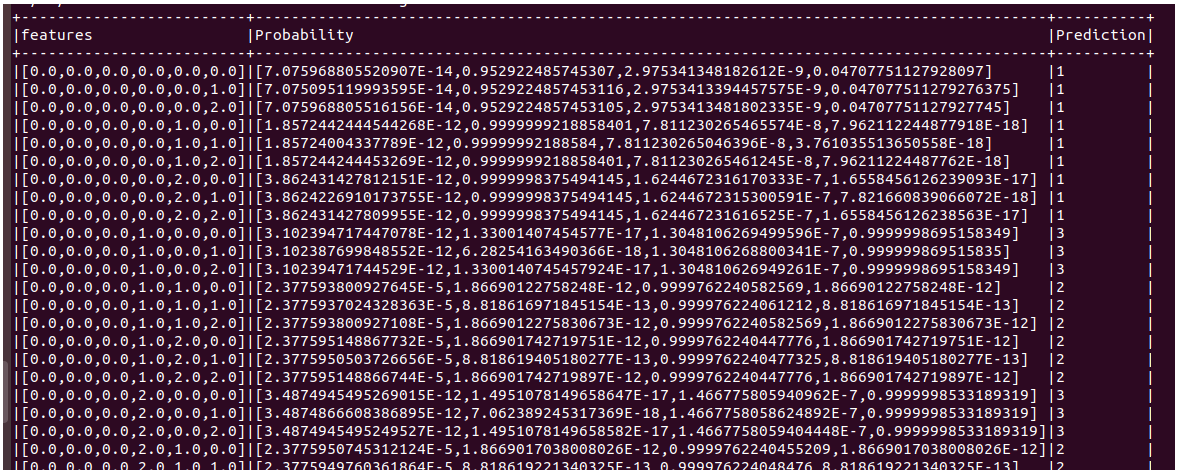
在该目录下，将之前的 spark.py 代码提交 spark 运行（根据你的存放路径而定）

```
./spark-submit /usr/local/experiment/GMM_Spark.py
```

之后运行过程会有很多输出。



最终在终端打印特征、可能性、预测结果的表格。



拓展实验（非必选）

Hadoop HDFS

本实验直接从磁盘上读取了数据。当数据非常庞大时，可能需要使用 Hadoop HDFS 管理。例如，你可以

将 car.txt 存到 Hadoop HDFS 中，然后从 Hadoop HDFS 中读取：

```
data = sc.textFile("hdfs://YOUR_IP_ADD:YOUR_PORT/user/root/data/car.txt")
```

Scala GMM

实验环节使用的是 python spark 相配合，进行 GMM 实验。事实上，Scala 是 Spark 的主要编程语言，旨在以简练、优雅的方式来表达常用编程模式。

你能否用 Scala 实现 GMM 并进行实验?

可参考 *GMM_Scala 实验手册.pdf*

Spark 独立部署模式（Standalone）

实验环节实际上是在 Spark Local 模式完成的，除此之外在生产环境中 Spark 还有独立部署模式和基于资源管理框架 Yarn 的 Yarn 模式。后两者都通过分布式存储和计算提高了生产效率。

你能否利用虚拟机的多开模拟实现 Spark 的独立部署模式并在集群之上完成任务提交？并对比 Local 模式和独立部署模式的完成时间?