实验七： Spark MLlib 库编程实践

实验基本信息：

时间：

实验类型： □验证性 □设计性 ☑综合性

班级： 计算机科学与技术专业（中外合作）4 班 班级代码:
理论老师： 邱开金 实验指导： 邱开金

实验报告提交说明：

本次实验需要撰写实验报告，实验报告填写时以完成实验任务为目的，先简要回答完成实验任务需要的步骤或需要执行的命令代码，再配以结果截图予以说明，图片数量不宜过多，能说明问题即可。最后配上总结，并提交到教师指定位置。

实验目的：

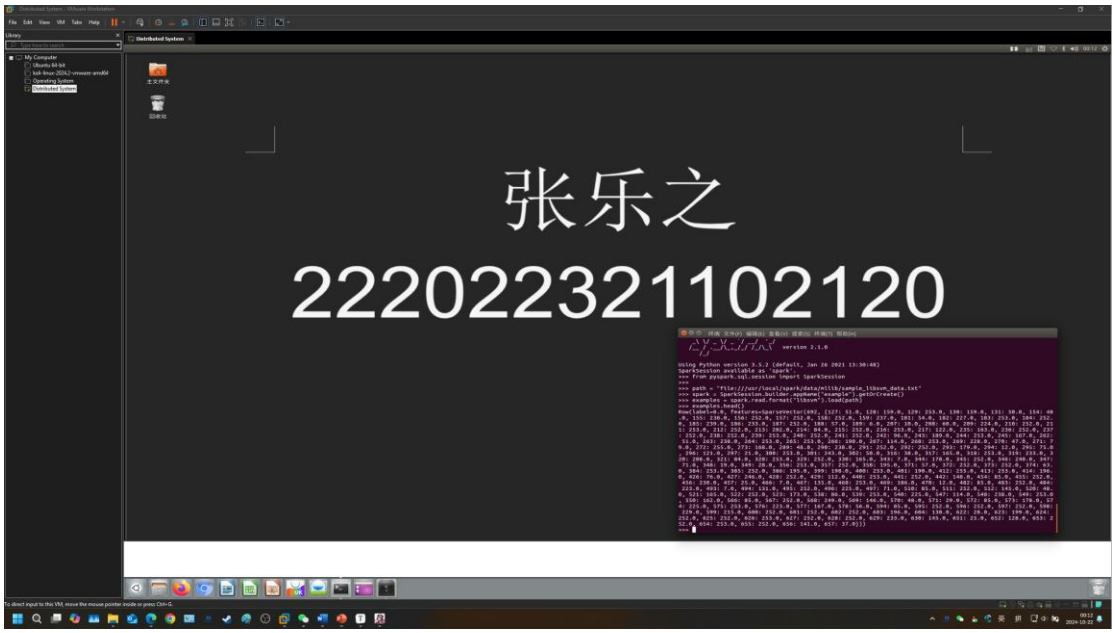
1. 了解 MLlib 包的主要特点。
2. 了解构建机器学习算法主要流程。
3. 熟练应用分类算法到具体数据集上，完成实验任务
4. 了解聚类算法
5. 了解机器学习算法评估

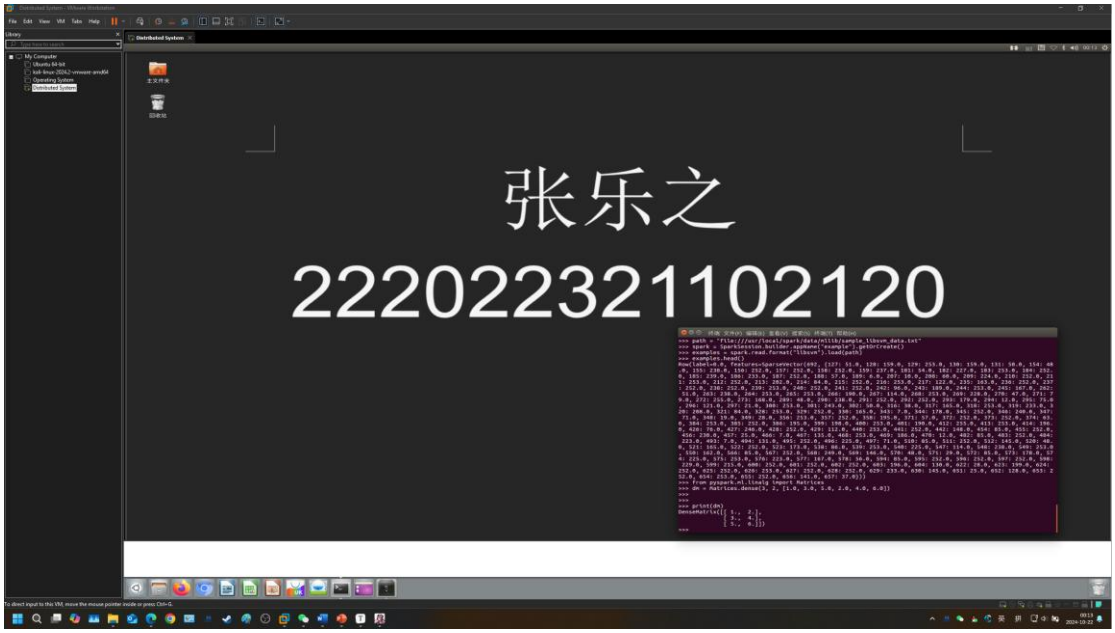
实验任务：

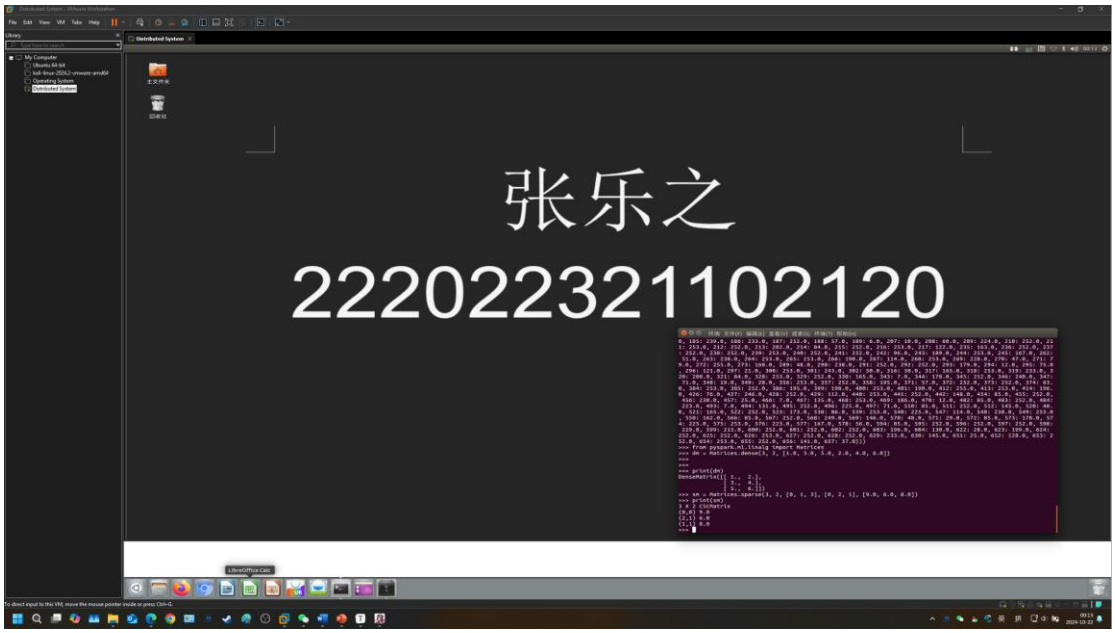
（作答要求：在每一个问题后写上所需要的命令，如果命令已经能够清楚回答问题，则不需要抓图。在所有命令执行完毕后，在终端输入 history 命令，将该命令执行结果抓图（抓图应能看清命令以及学生所用的电脑的当前时间或当前桌面背景等能区分出是自己做的证据信息）附在最后）

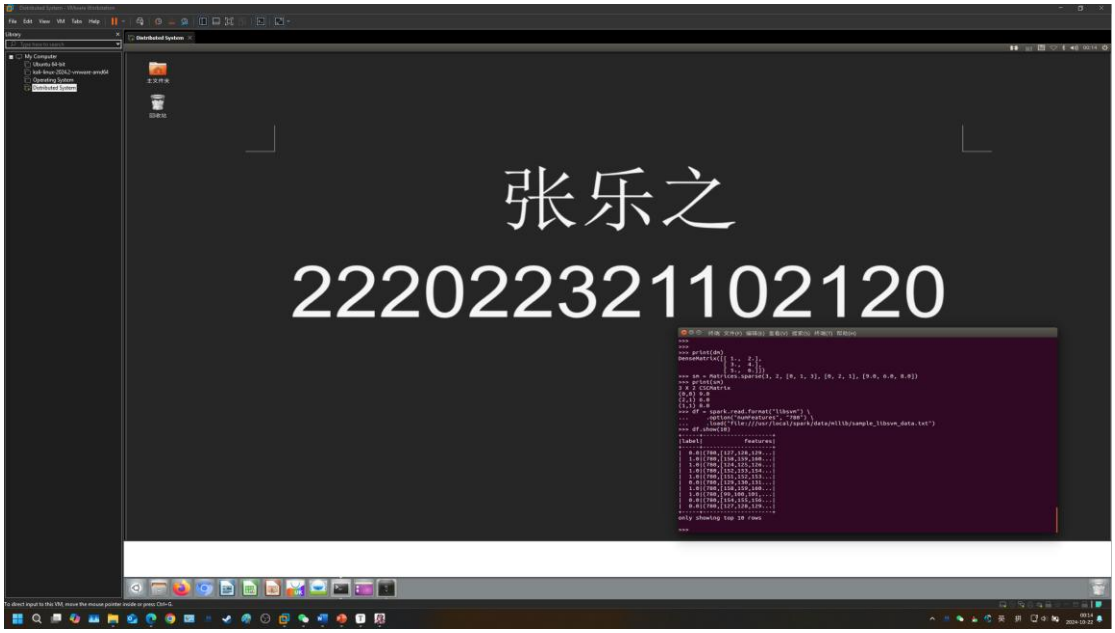
1. 完成 PPT 第 12-48 页黑框内命令，完成后截图。

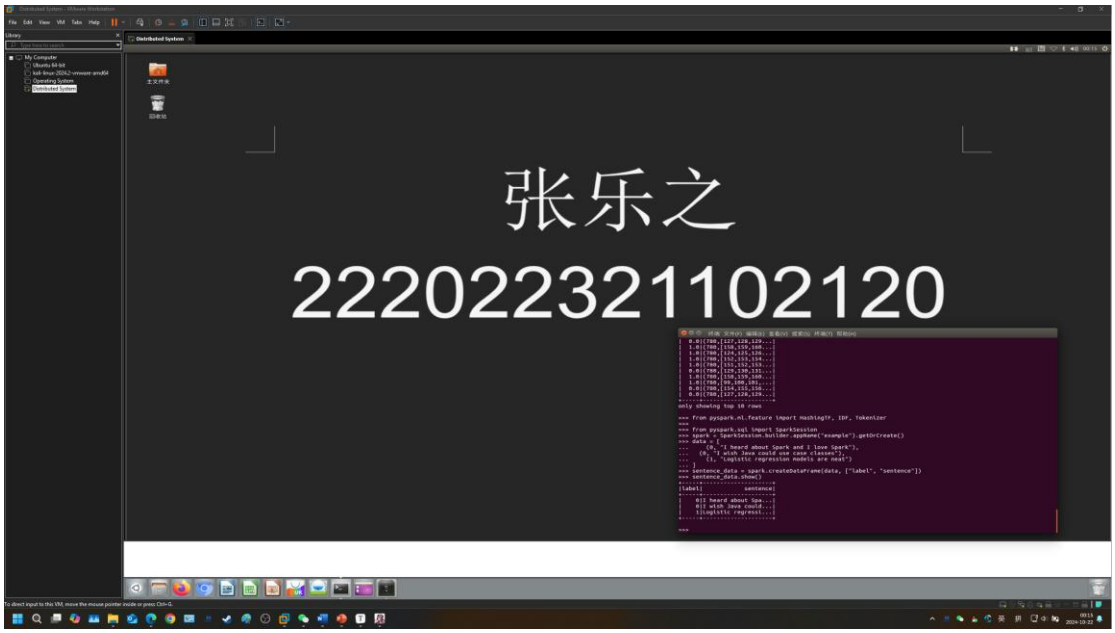

张乐之
222022321102120
Using Python version 3.5.2 (default, Jan 26 2021 13:30:48)
SparkSession available as 'spark'.
>>> from pyspark.ml.linalg import Vectors
>>> dv = Vectors.dense([2.0, 0.0, 8.0])
>>> sv1 = Vectors.sparse(3, [0, 2], [2.0, 8.0])
>>> sv2 = Vectors.sparse(3, [(0, 2.0), (2, 8.0)])
>>> print("稠密向量 dv:", dv)
稠密向量 dv: [2.0,0.0,8.0]
>>> print("稀疏向量 sv1:", sv1)
稀疏向量 sv1: (3,[0,2],[2.0,8.0])
>>> print("稀疏向量 sv2:", sv2)
稀疏向量 sv2: (3,[0,2],[2.0,8.0])
>>> from pyspark.mllib.regression import LabeledPoint
>>> from pyspark.mllib.linalg import Vectors
>>> from pyspark.sql.session import SparkSession
>>> pos = LabeledPoint(1.0, Vectors.dense([2.0, 0.0, 8.0]))
>>> neg = LabeledPoint(0.0, Vectors.sparse(3, [0, 2], [2.0, 8.0]))
>>> print("Positive Example:", pos)
Positive Example: (1.0,[2.0,0.0,8.0])
>>> print("Negative Example:", neg)
Negative Example: (0.0,(3,[0,2],[2.0,8.0]))
>>>
[1]+ 已停止 bin/pyspark
hadoop@hadoop-virtual-machine:/usr/local/spark$

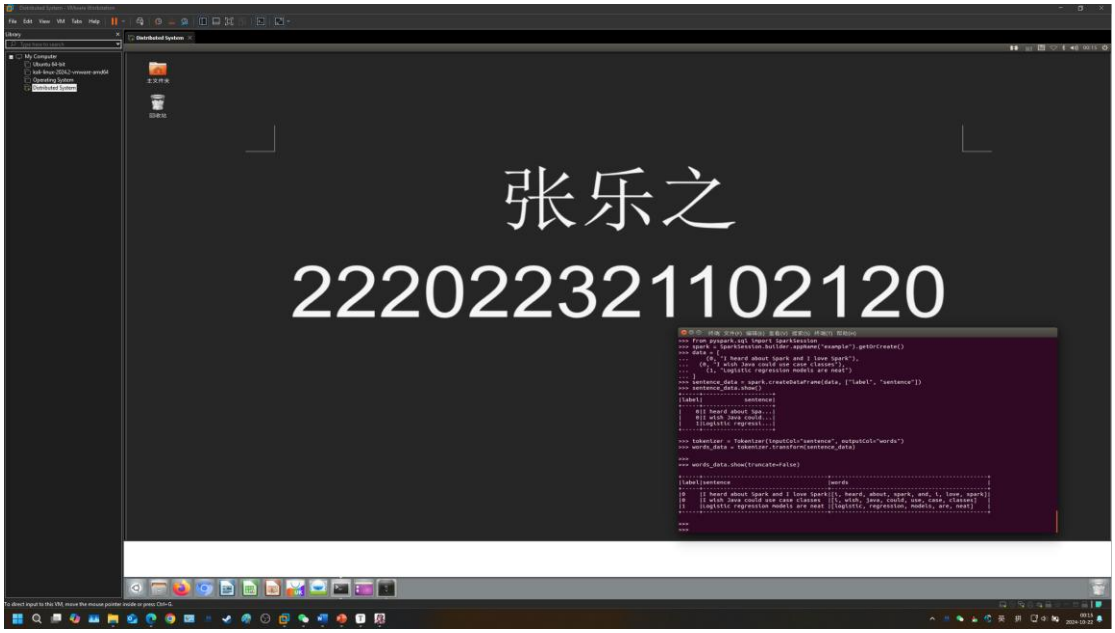

张乐之
222022321102120

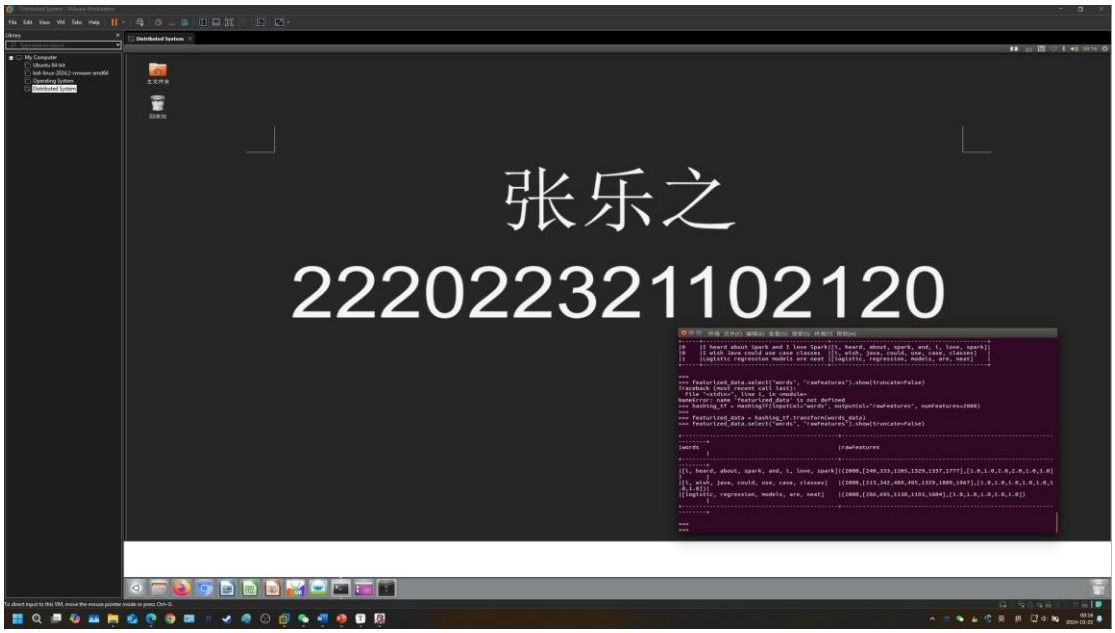

张乐之
222022321102120

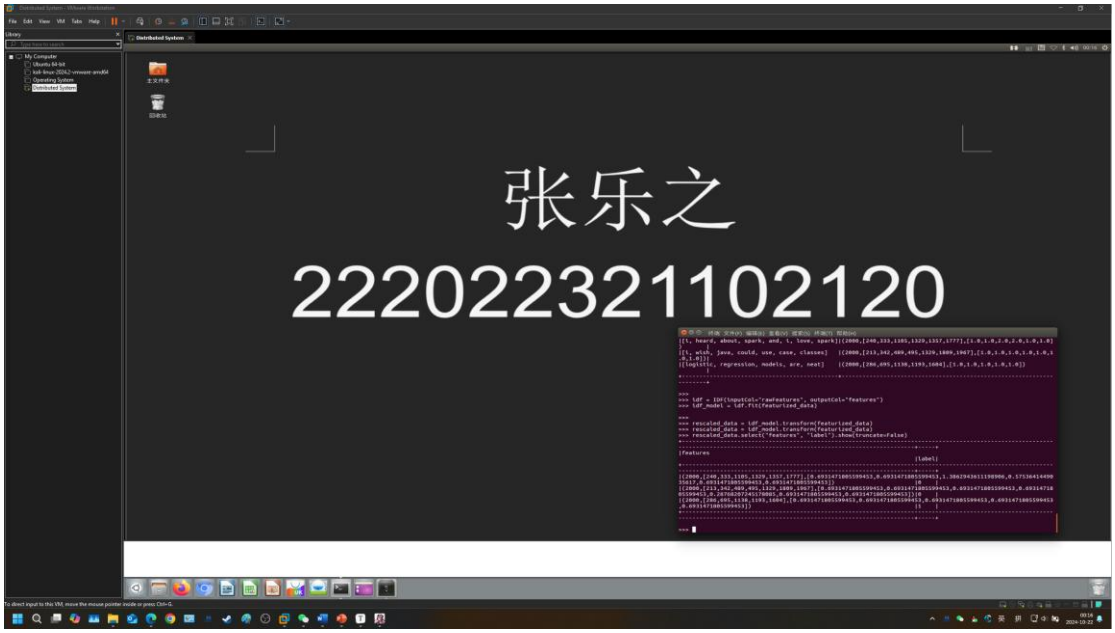

张乐之
222022321102120


张乐之
222022321102120


张乐之
222022321102120


张乐之
222022321102120


张乐之
222022321102120



2. iris.txt 鸢尾花数据集介绍

```
SepalLength,SepalWidth,PetalLength,PetalWidth,TrainingClass
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
5.3,3.7,1.5,0.2,Iris-setosa
5.0,3.3,1.4,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
5.5,2.4,3.7,1.0,Iris-versicolor
5.8,2.7,3.9,1.2,Iris-versicolor
6.3,3.3,6.0,2.5,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
6.2,3.4,5.4,2.3,Iris-virginica
5.9,3.0,5.1,1.8,Iris-virginica
```

4 个属性分别是花萼（读 e 四声）长，花萼宽，花瓣长，花瓣宽，花的类别。其中花的类别共有 3 类，分别是 Iris-setosa，Iris-versicolor，Iris-virginica。

2. iris.txt 利用 SVM 进行二分类（**根据下方代码，在自己机器上操作一遍**）

1) 数据准备：去除 iris.txt 中的标题行

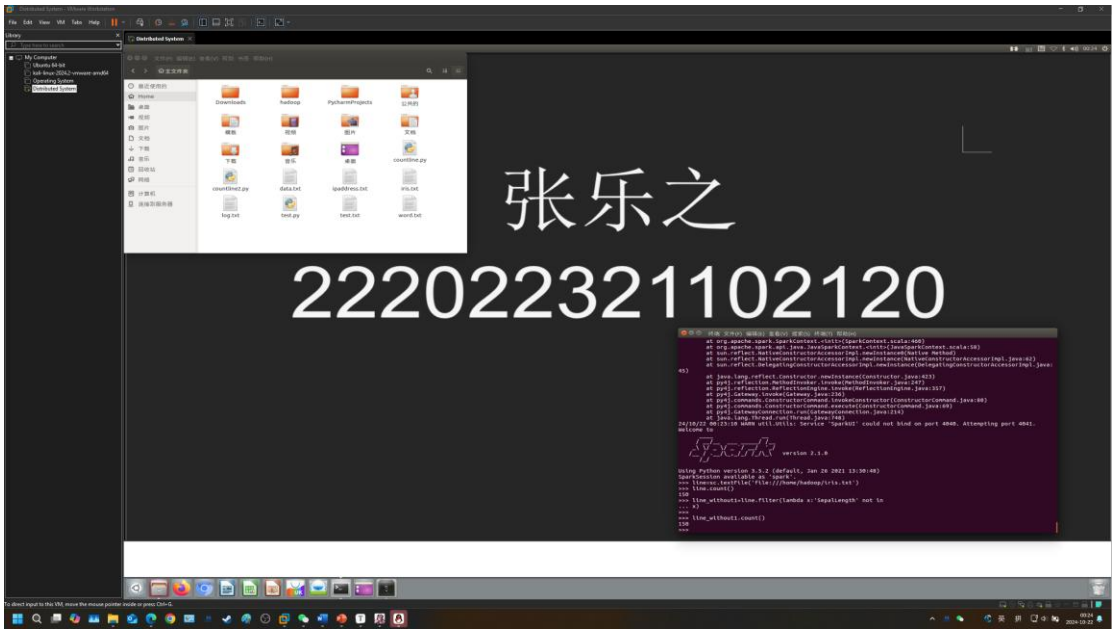
启动 pyspark

在命令行输入 pyspark

读入数据/home/Hadoop/iris.txt,显示有多少行，并去除标题行

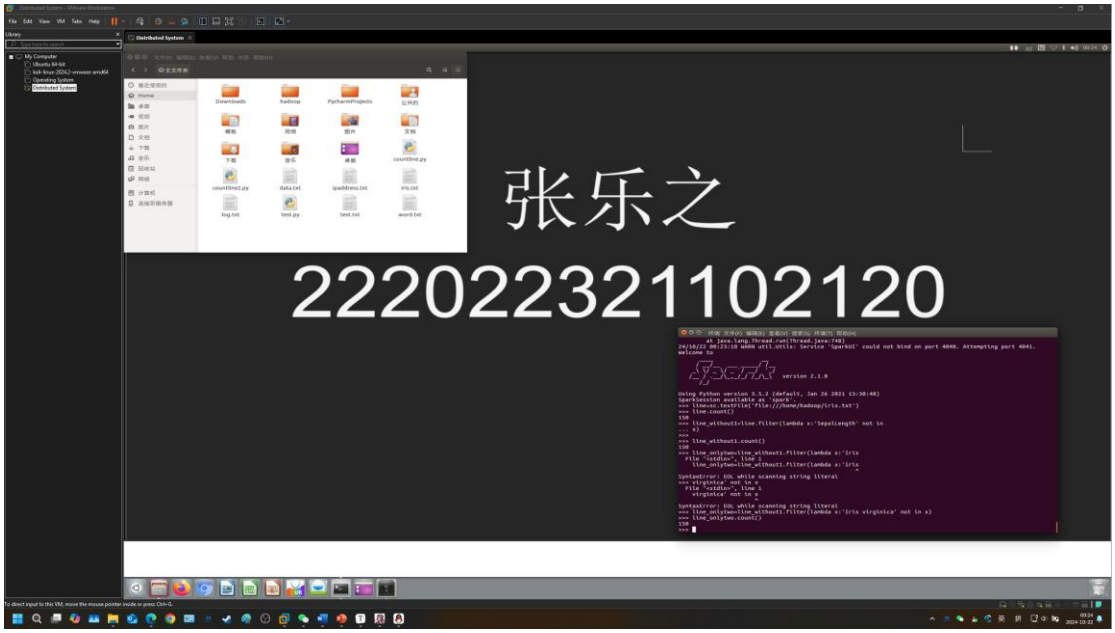
```
>>> line=sc.textFile('file:///home/hadoop/iris.txt')
>>> line.count()
151
>>> line_without1=line.filter(lambda x:'SepalLength' not in
x)
>>> line_without1.count()
```

150



2）数据准备：去除 iris.txt 中 Iris-virginica 这类数据

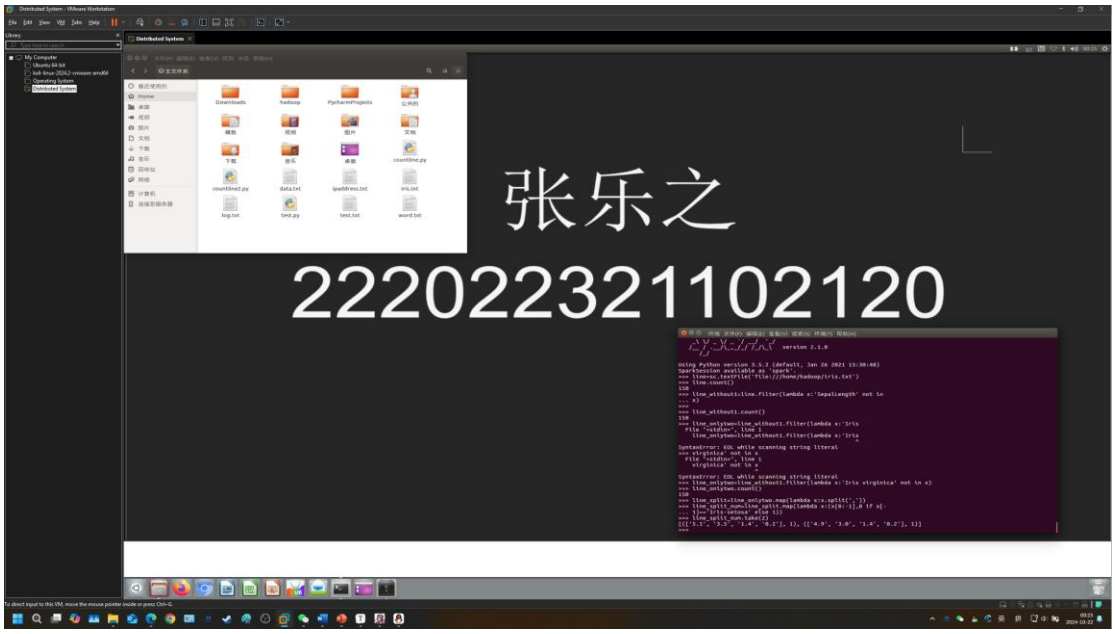
```
>>> line_onlytwo=line_without1.filter(lambda x:'Iris virginica' not in x)
>>> line_onlytwo.count()100
```



3）数据准备：将 iris.txt 中类别换成 0 和 1，Iris-setosa 对应为 0，Iris-versicolor 对应为 1.

```
>>> line_split=line_onlytwo.map(lambda x:x.split(','))
>>> line_split_num=line_split.map(lambda x:(x[0:-1],0 if x[-1]=='Iris-setosa' else 1))
>>> line_split_num.take(2)
[(['5.1', '3.5', '1.4', '0.2'], 0), (['4.9', '3.0', '1.4',
```

'0.2'], 0)]



4) 构建分类算法需要的 LabeledPoint 格式

导入需要的包

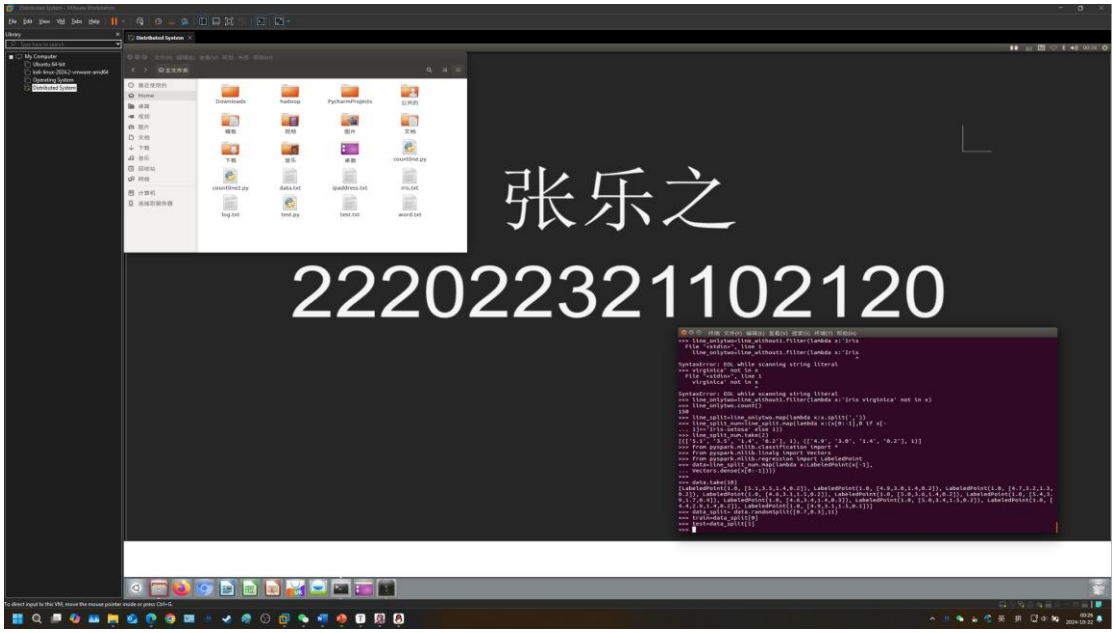
```
>>> from pyspark.mllib.classification import *
>>> from pyspark.mllib.linalg import Vectors
>>> from pyspark.mllib.regression import LabeledPoint
```

转换数据格式为 LabeledPoint

```
>>>
data=line_split_num.map(lambda x:LabeledPoint(x[-1],
Vectors.dense(x[0:-1])))
>>> data.take(10)
```

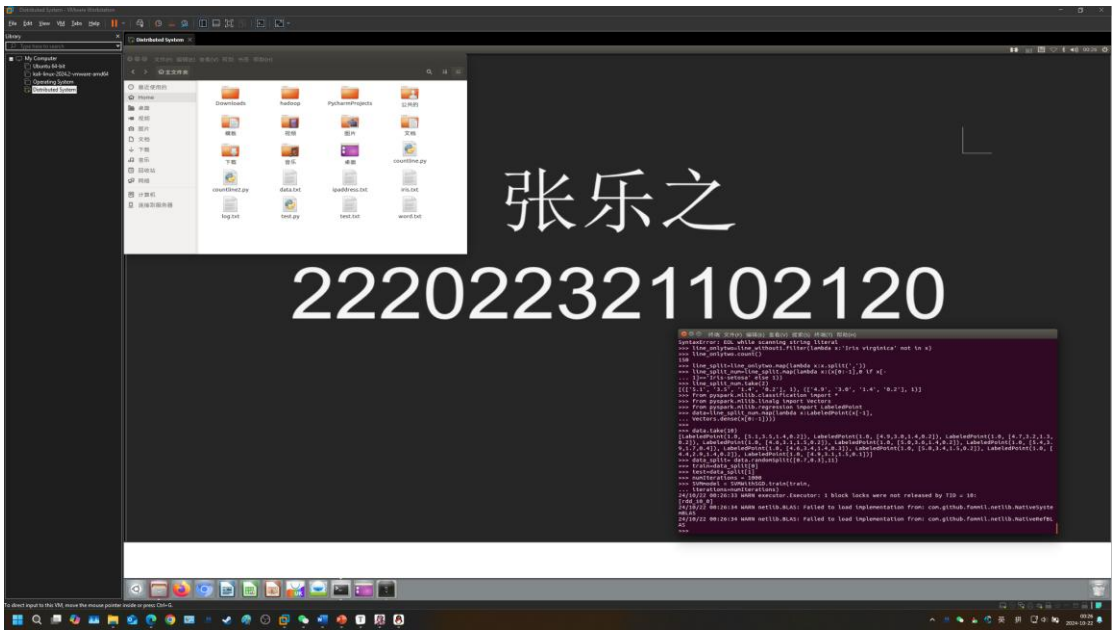
拆分数据为训练集和测试集

```
>>> data_split= data.randomSplit([0.7,0.3],11)
>>> train=data_split[0]
>>> test=data_split[1]
```



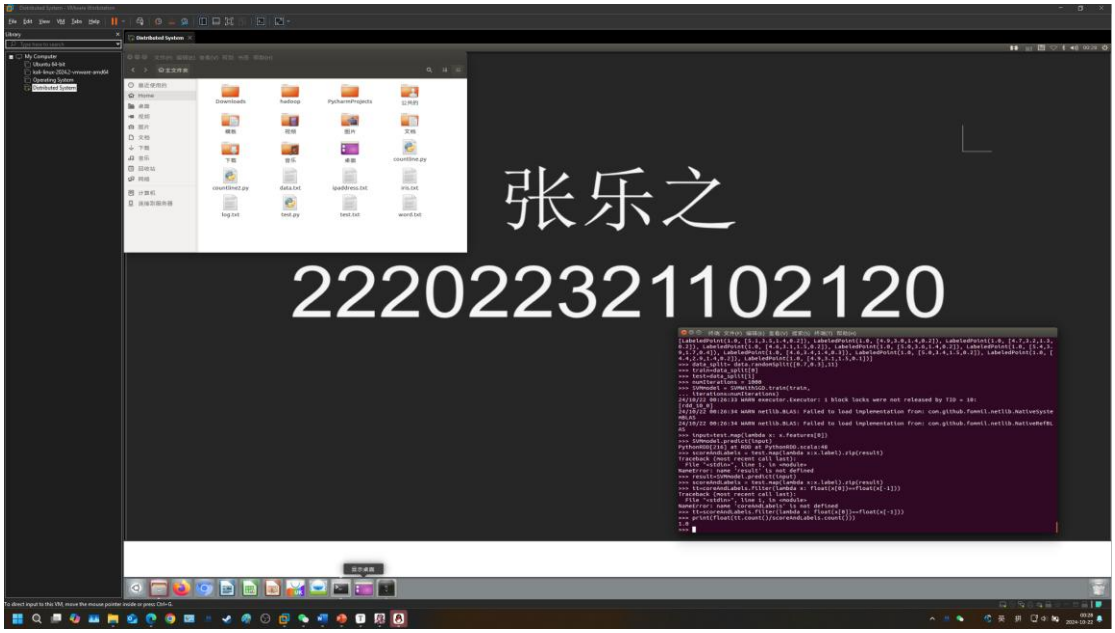
5）运行分类算法

```
>>>numIterations = 1000
>>>SVMmodel = SVMWithSGD.train(train,
iterations=numIterations)
```



6）评估分类结果

```
# 如下方法计算结果>>>input=test.map(lambda x: x.features[0])
>>>result=SVMmodel.predict(input)
#下面这几句合成类标和预测值
>>>scoreAndLabels = test.map(lambda x:x.label).zip(result)
#计算分类正确的比例
>>>tt=scoreAndLabels.filter(lambda x: float(x[0])==float(x[-1]))
>>>print(float(tt.count()/scoreAndLabels.count()))
```



3. 查找 pyspark 官方文档，**利用 RandomForest 算法完成多分类**

1）核心代码提示：把三个分类名称变成数字

```
def change_class(x):
if x==’Iris-setosa’:
return 0
if x==’Iris-versicolor’:
return 1
else:
return 2
```

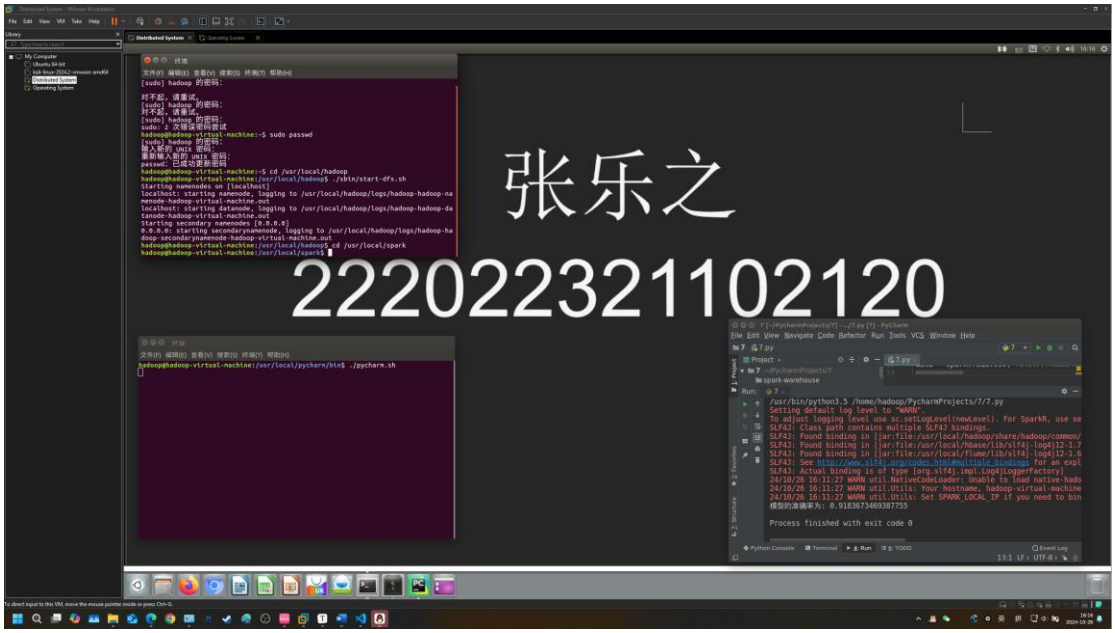
spark 中这样调用即可

```
line_split_num=line_split.map(lambda x:(x[0:-
1],change_class(x[-1])))
```

代码如下：

```
from pyspark.sql import SparkSession
from pyspark.ml.feature import StringIndexer, VectorAssembler
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.sql.functions import udf
from pyspark.sql.types import IntegerType

# 创建 SparkSession
spark =
SparkSession.builder.appName("RandomForestMultiClassification").getOr
Create()

# 加载数据集（假设数据存储在 CSV 文件中）
data = spark.read.csv("file:///home/hadoop/iris.txt", header=True,
inferSchema=True)
```

```
# 定义函数将分类标签转换为数字
def change_class(label):
      if label == 'Iris-setosa':
            return 0
      elif label == 'Iris-versicolor':
            return 1
      else:
            return 2

# 使用 UDF 将分类标签转换为数字
change_class_udf = udf(change_class, IntegerType())
data = data.withColumn("TrainingClass",
change_class_udf(data["TrainingClass"]))

# 使用 VectorAssembler 将特征列组合成一个名为 'features' 的向量列
assembler = VectorAssembler(inputCols=['SepalLength', 'SepalWidth',
'PetalLength', 'PetalWidth'], outputCol="features")
assembled_data = assembler.transform(data)

# 划分训练集和测试集
train_data, test_data = assembled_data.randomSplit([0.7, 0.3])

# 创建 RandomForest 分类器
rf = RandomForestClassifier(labelCol="TrainingClass",
featuresCol="features", numTrees=10)

# 训练模型
model = rf.fit(train_data)

# 进行预测
predictions = model.transform(test_data)

# 模型评估
evaluator = MulticlassClassificationEvaluator(labelCol="TrainingClass",
predictionCol="prediction", metricName="accuracy")
accuracy = evaluator.evaluate(predictions)

print("模型的准确率为: {}".format(accuracy))
```



2) RandomForest 算法调用说明
http://spark.apache.org/docs/2.1.1/api/python/pyspark.mllib.html#module-pyspark.mllib.tree4.

参考资料及提示：

1.
转换成 labeledpoint,提示引入包，numpy 未安装，需要安装
sudo apt install python3-pip
pip3 install -i https://mirrors.aliyun.com/pypi/simple/ numpy
2.
Spark MLlib 简介 http://dblab.xmu.edu.cn/blog/1762-2/
3.
Spark 官方文档 http://spark.apache.org/docs/2.1.1/api/python/index.html

实验总结：

在本次实验中，我主要完成了基于 Spark 的 RandomForest 算法多分类模型的实现与可视化，探索了 Spark MLlib 在分类任务中的应用，尤其是 RandomForest 的模型训练、预测及其在 Spark 上的应用流程。
首先，我编写了数据预处理代码，将分类标签转化为数值编码。通过定义 change_class 函数，成功将数据集中类别名称（如 Iris-setosa、Iris-versicolor 和 Iris-virginica）分别转化为 0、1 和 2，方便后续模型处理。同时，通过 Spark 的 VectorAssembler 将多列特征组合为单一的 features 向量列，为模型训练提供了标准格式的输入数据。
接下来，我基于 Spark MLlib 实现了 RandomForest 多分类模型的训练与预测。借助 RandomForestClassifier，在训练集上成功训练出分类模型，并在测试集

上进行了预测。通过 MulticlassClassificationEvaluator 计算了模型在测试集上的准确率，初步验证了模型的分类性能。
实验的最后一部分，我将预测结果导出至 Pandas 数据框，利用 Matplotlib 和 Seaborn 进行可视化。通过绘制散点图和混淆矩阵，我直观展示了模型的分类效果，其中散点图显示了真实类别和预测类别的分布情况，而混淆矩阵则帮助分析了分类结果的准确性，揭示了模型在不同类别上的表现差异。
总体而言，此次实验让我深入了解了 Spark 在机器学习中的应用，尤其是 RandomForest 在大数据平台上的实现方法。同时，通过数据可视化，我进一步掌握了如何解读模型预测结果，从而为后续的模型优化提供了基础。这次实验不仅提升了我对 Spark 和 MLlib 的理解，也为今后更复杂的机器学习任务打下了坚实的基础。