Part II of Project of Spark

Student\_ID:16307130214 Name:Liu Baichuan

1. Introduction

* Tasks of this part of Spark Project is using RDD operation to implement some complex statistical work and using pyspark machine package to do some ML tasks. In this part, I learn how to use pyspark machine package and know the difference between ml and mllib.
* My Project runs on Ubuntu16.04 + Spark2.4 + Hadoop2.7 as platform.
* My code firstly writes and runs on Ipython notebook, so it is more convenient for me to paste printscreens in this report. More important, it’s my fault to forget to record my code. So in my commit files, you have to run my code in notebook line by line and do not run it as python script in pyspark. I also know if I want to use python script, I have to use SparkConf and SparkContext which are loaded automatically in Ipython. But you can run h1 to h3 by ./spark-submit
* Potential application: Have the ability to use large scale distributed system.
* Spark is a great platform to process big data. According to its document[[1]](#footnote-1),

Apache Spark is a fast and general-purpose cluster computing system. It

provides high-level APIs in Java, Scala, Python and R, and an optimized

engine that supports general execution graphs. It also supports a rich set of

higher-level tools including Spark SQL for SQL and structured data processing,

MLlib for machine learning, GraphX for graph processing, and Spark

Streaming.

1. Q and A

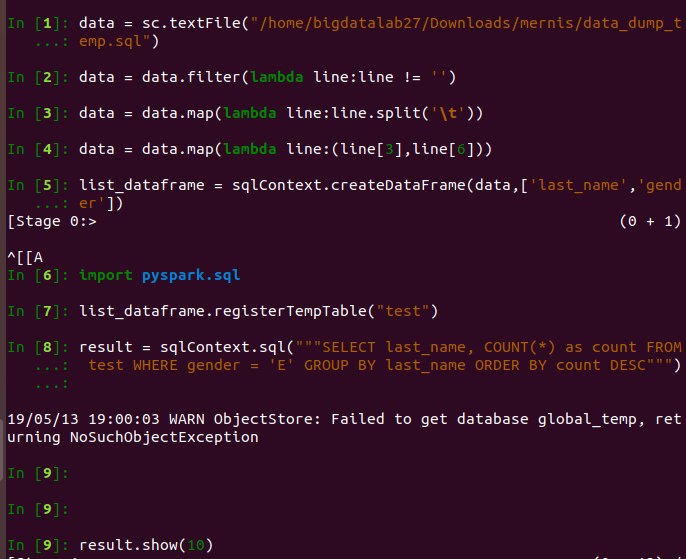
N6. 统计男性、女性中最常用的5个名字

Algorithm：

* Extract name(first name + last name) and gender information
* Register a table
* Use sql GroupBy and OrderBy to find the most common last name respectively

Process:

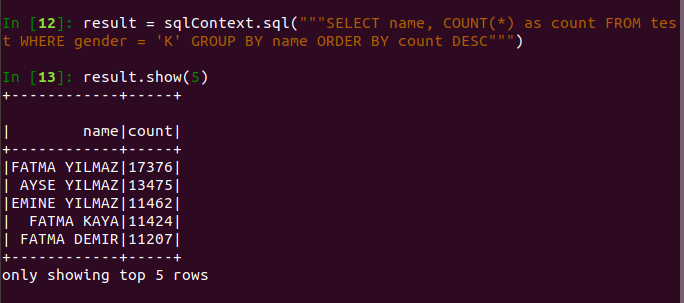
name and gender field



****

Register table and use sql

****

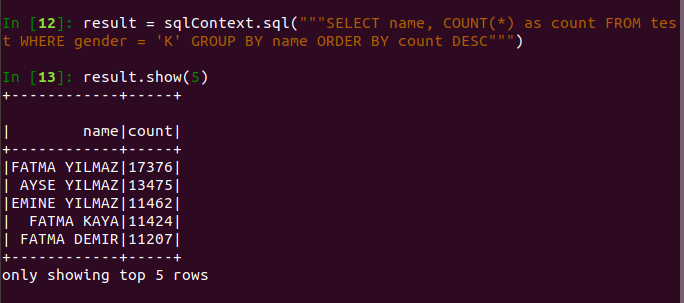
****

Result:

male

****

female

****

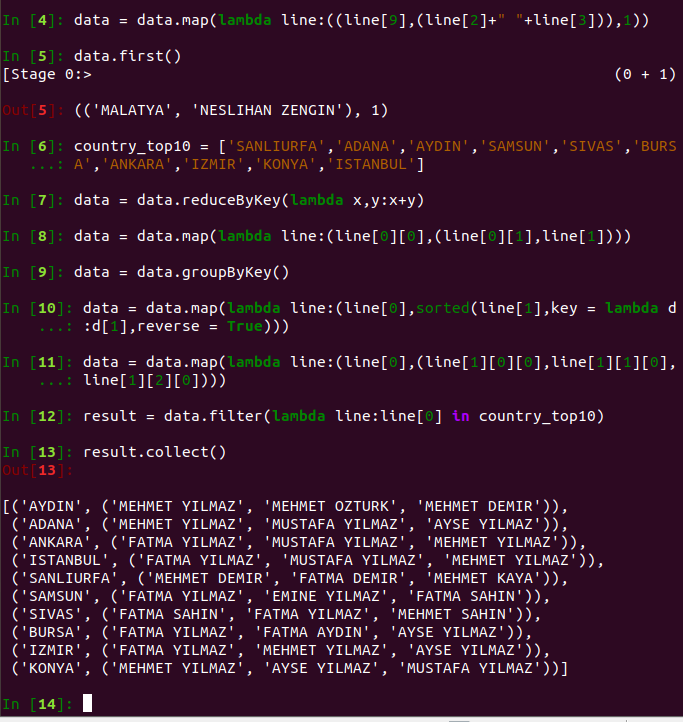
It is easy to find that MEHMET is the most popular name among male while FATMA is the most popular name among female.

N7. 统计一下该国前10大人口城市中，每个城市最受欢迎的3个名字是什么 Algorithm：

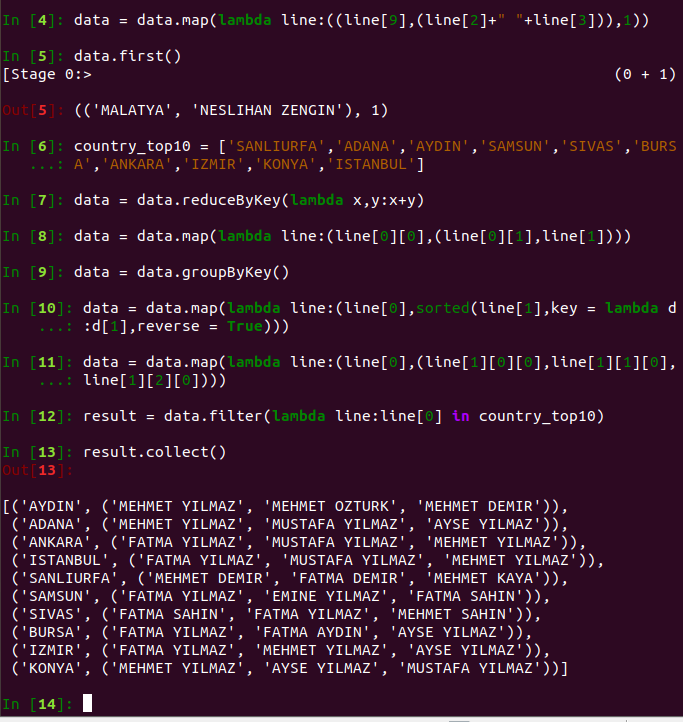
* Create RDD pair((city,name),1) and count by key
* Change RDD pair ((city, name),count) to (city,(name,count))
* Merge RDD pair by key and sort RDD pair by count in value
* Extract top3 name for top 10 population city(find in N4)

Process:

Get rdd (city,(name,count))

****

Result:

****

We may find popular names differ from city to city.

In machine learning(Hard part), I will use Naïve Bayes in machine learning packages of pyspark. In ml and mllib, many classification method only applies for two-category classification. Decision Tree and Naïve Bayes apply for multi-category classification. I do think I will not choose more than 3 features to train model, so Decision Tree is not a good idea. Actually, three machine learning tasks in Hard part ask me to split data as train data, valid data, test data. However, valid data is not so useful for improving model, so I will not use valid for K-cross validation. Tasks ask me to compute topN accuracy. Mllib only provide prediction label while ml provide predict probability, so I have to use ml package. You will see my attempt in H1. Also, features are designed as discrete variable. I will not use ont-hot encode because one feature has a large of variables.

H1. 构建人所在的城市的预测模型：根据给定一个人的所有信息（除了所在城市），预测该人所在的城市。分析该模型Top1到Top5的准确率

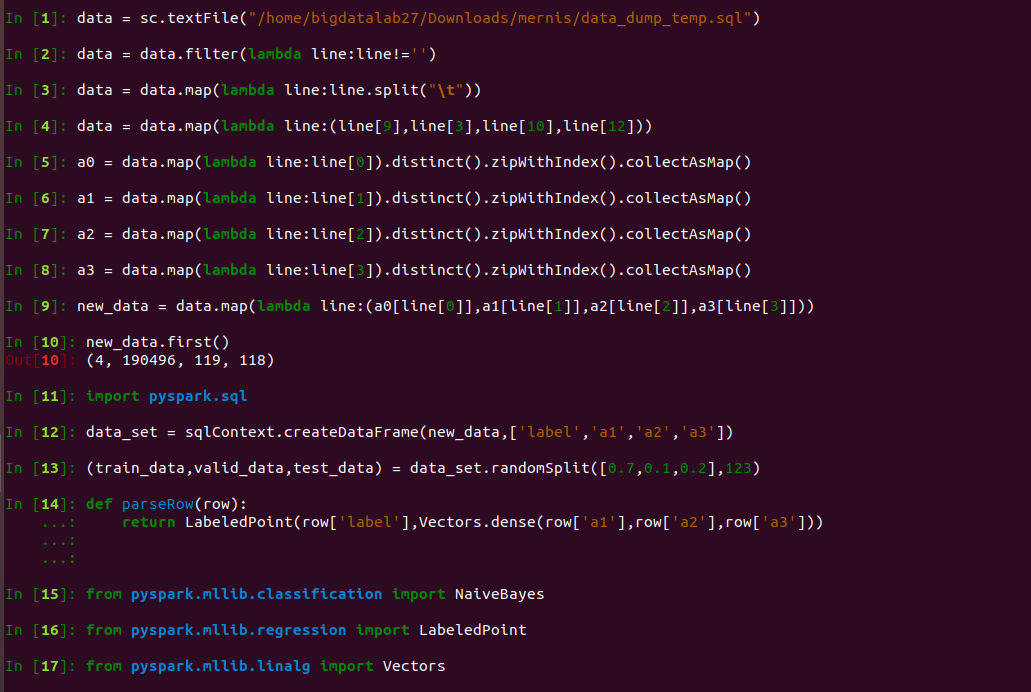
Algorithm：

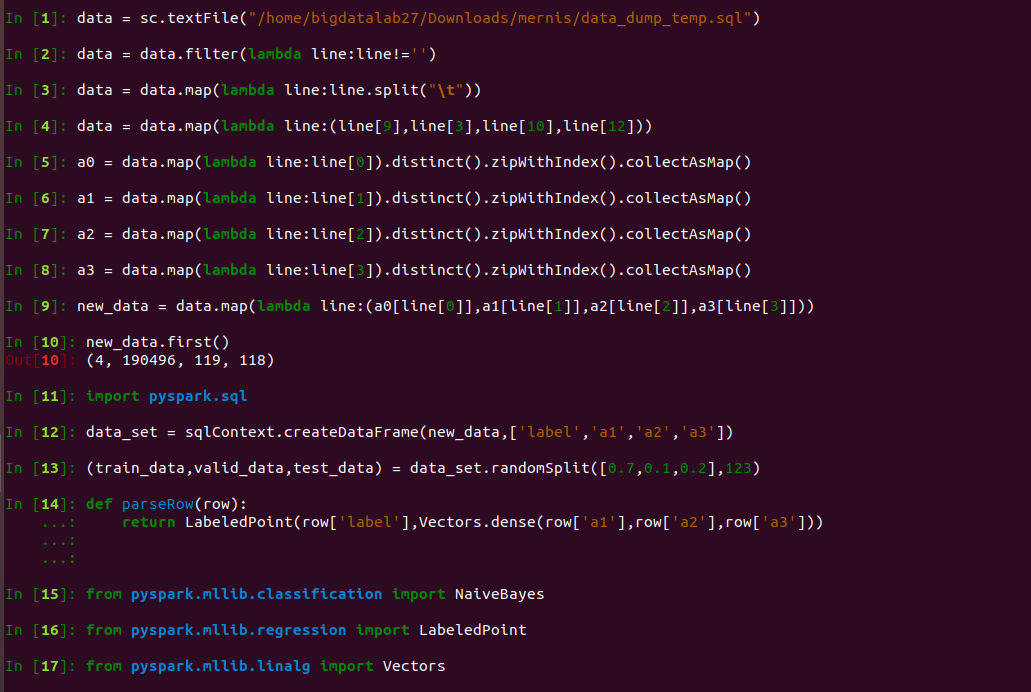
* Choose features
* Change data to specific type which is suitable for ml or mllib
* Use train data to train model
* Use test data to compute accuracy

Process:

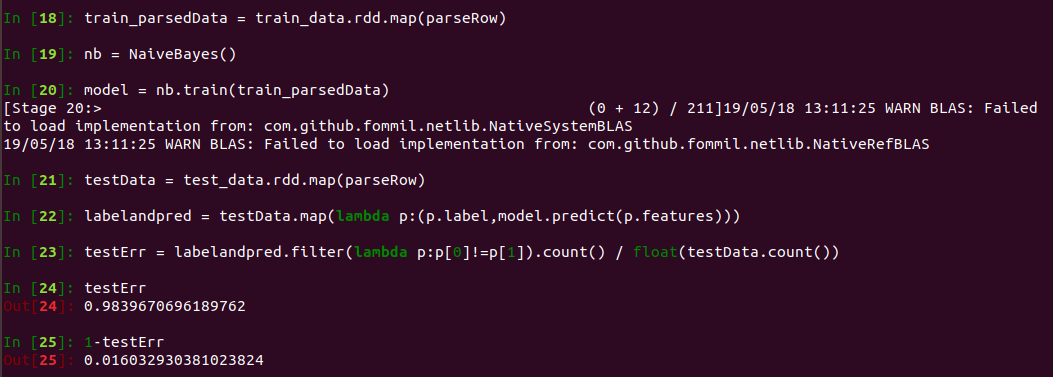
At beginning, I misunderstand this task. I think I can not choose any feature which is about on city, so I choose last\_name, id\_registration\_district, address\_district as features.(Also choose a wrong label: id\_registration\_city, correct label should be address\_city)

Use zipWithIndex to transform text features to digital features

****

****

Definite a parseRow function to construct my label and features as LabeledPoint type which is suitable for mllib. Split the dataset.

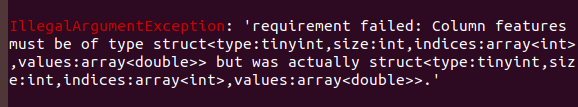
****

Obviously, this model is horrible. So next I will change features as birth\_city,id\_registration\_city. Then I also find mllib can not compute topN. I use ml next. Ml and mllib ask for different data type. At first, I only change my parseRow function.

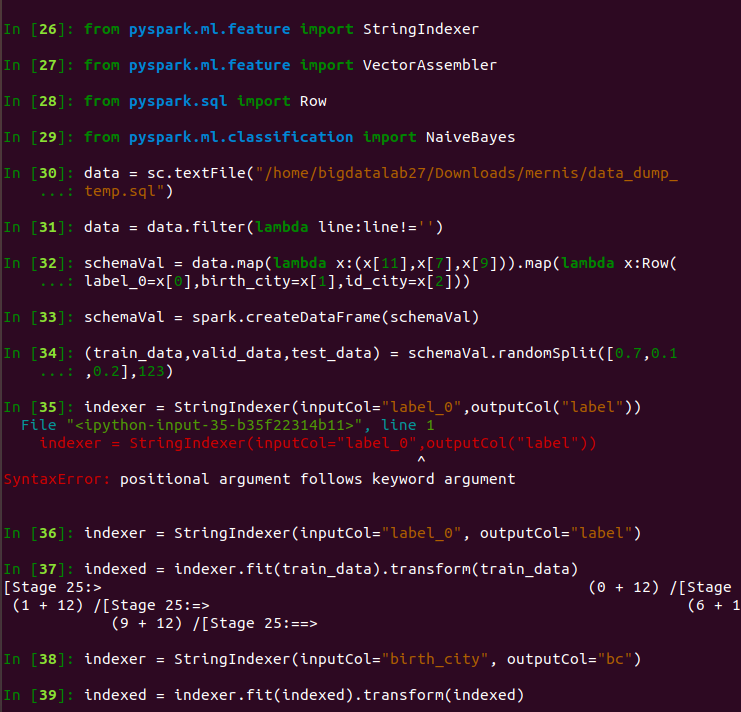
def parseRow(row):

return Row(row[‘label’],Vectors.dense(row[‘a1’],row[‘a2’],row[‘a3’]))

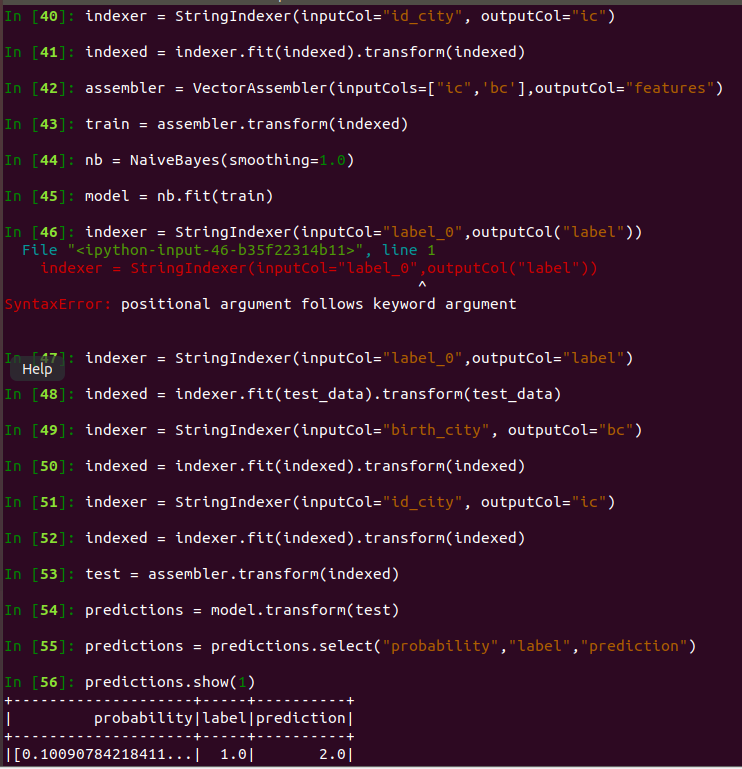
However, I always get this error which I can not solve.

****

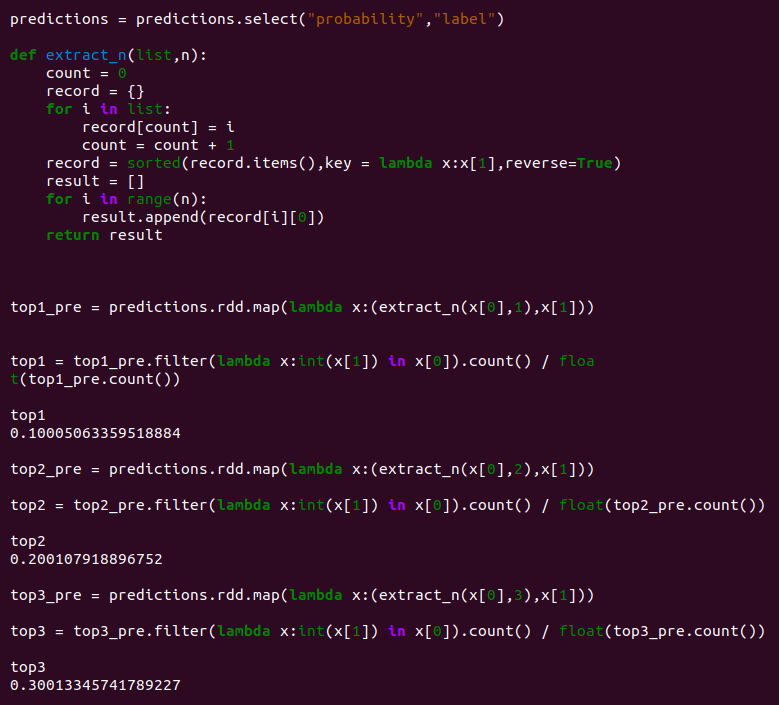
I then completely change my code by using StringIndexer, VectorAssembler for ml.

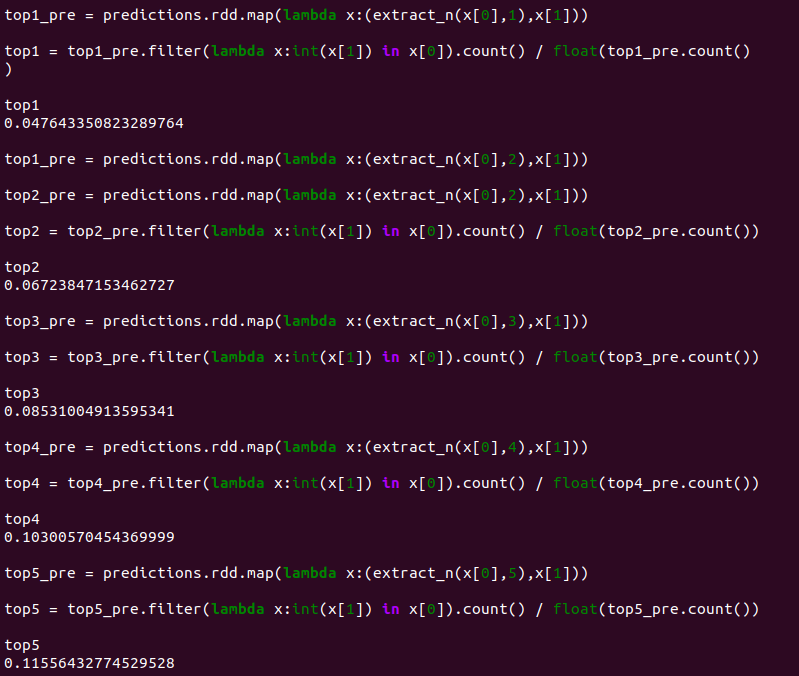
****

Choose features and StrngIndexer can transform text features to digital features.

****

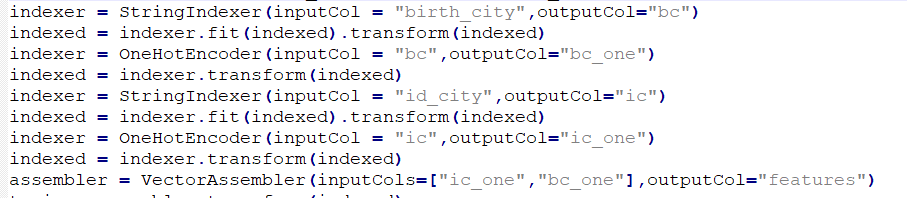
Use train data train Naïve Bayes model. Change test data to suitable struct and use model to predict. Then I find I can use probability for every label. So I can compute topN accuracy.

****



Top1 is 4.78%,top2 is 6.72%, top3 is 8.53%,top4 is 10.30% ,top5 is 11.56%.

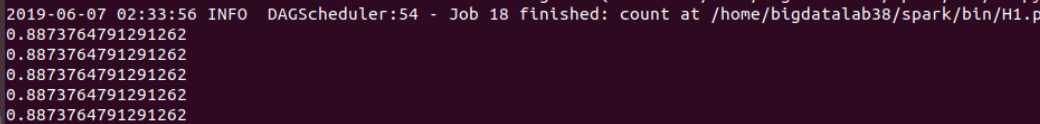
However, it is not good. Have to use One-hot-encoder, naïve bayes only can understand my feature. I change my code to understand features.



After modifying(also modify the size of dataset to 10000000) (refer to my submitted code), new top5:

(H2、H3 have same problem, the alters have been shown in code. And New TopN will be given directly)

I find a strange thing that Top1 – Top5 are same, which means if model does not predict the label, the correct prediction will not show up with low probability. However, my model actually has a high accuracy.



H2. 性别预测模型：根据给定的一个人的信息（除了性别），能否给出该人的性别

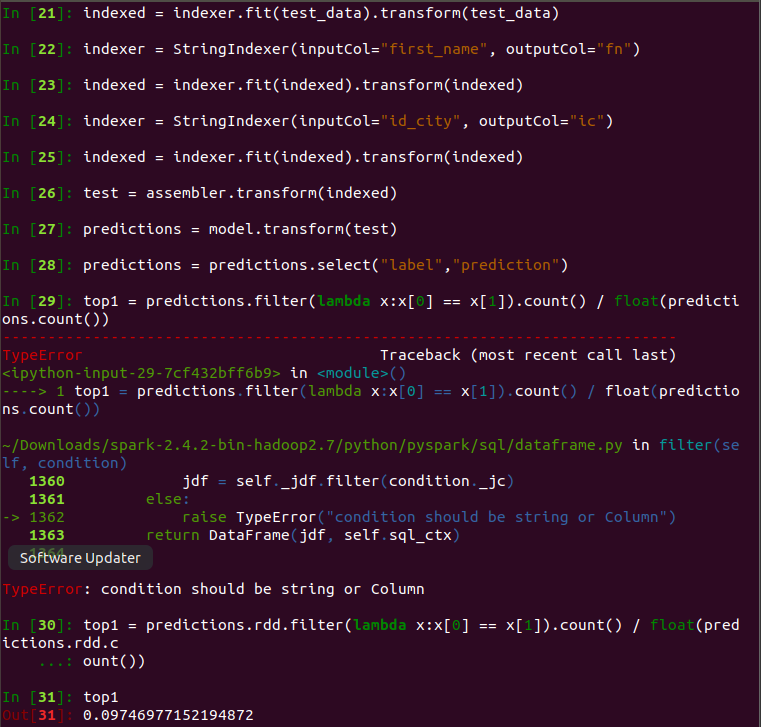
Algorithm：

* Choose features
* Change data to specific type which is suitable for ml or mllib
* Use train data to train model
* Use test data to compute accuracy

Process:

This tasks is similar to H1 except for label, features and TopN(because labels only have two different values)

I use name and id\_registration\_city as features.

**\**

The result is that the accuracy is 0.506.

New accuracy is 89.74%.

H3. 姓名预测模型：根据给定的一个人的信息（除了姓名），能否给出该人的姓氏。分析该模型TOP1 到TOP5的准确率。

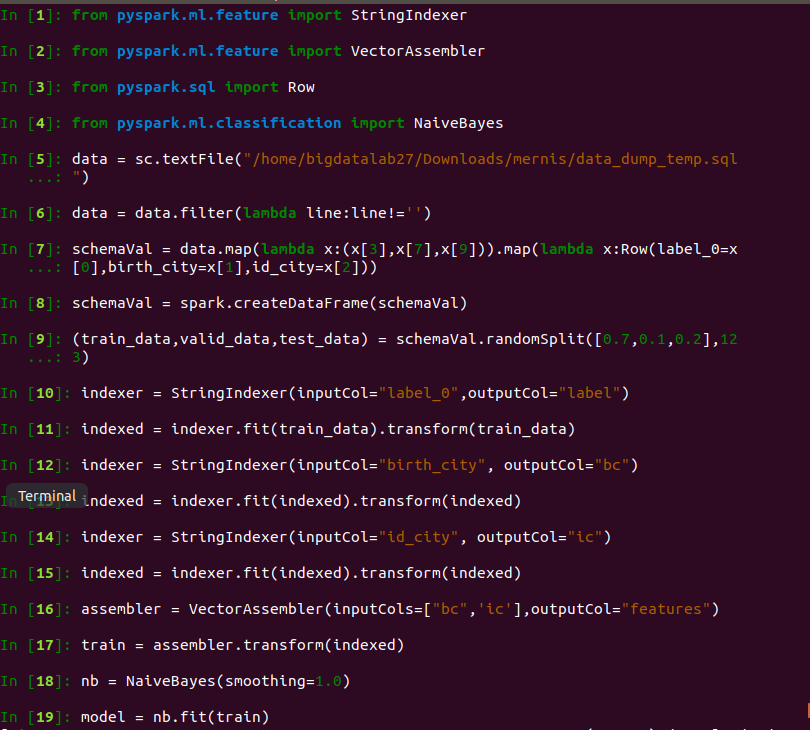
Algorithm：

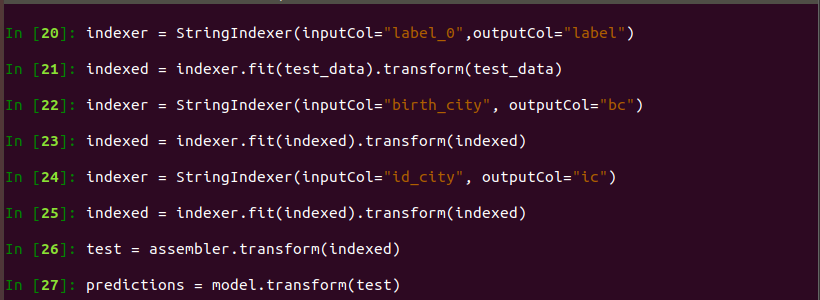
* Choose features
* Change data to specific type which is suitable for ml or mllib
* Use train data to train model
* Use test data to compute accuracy

Process:

This tasks is similar to H1 except for label.

Why I choose these two features? Hinted by statistics tasks, people in different cities may tend to choose different name. In other words, name is a regional character.

****

****

Only choose label and probability in predictions. Definite a function to extract topN predicted labels, if label shows in them, this prediction will be correct. Function has a parameter N to simplify TopN. I have to mention ml use data frame as data type, when use RDD operation , I have to use .rdd to change data frame to RDD.

As code in H1, I get top1 – top5 are 1.12%, 2.04%,2.68%,3.26%,3.64%.

New topN are 13.31%,15.32%,15.32%,15.32%,15.32%, accuracy is not still good. I think the reason is the number of the labels(last\_name) is too large to predict correctly.

1. http://spark.apache.org/docs/latest/ [↑](#footnote-ref-1)