

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**



LAB – 03:

Spark

Lớp: 19_21

Môn: Nhập môn dữ liệu lớn

Niên khóa: 2021-2022

Mục lục

I. Thông tin sinh viên	3
II. Nội dung tìm hiểu	4
1. Cài đặt Spark trên Linux:	4
2. Cài đặt thuật toán và thực hiện chạy các ví dụ và bộ dữ liệu khác:	11
III. Tài liệu tham khảo	18

I. Thông tin sinh viên

1. Thông tin nhóm:

Tên nhóm: Gaming House.

Danh sách thành viên:

STT	Họ tên	MSSV
1	Đỗ Thái Duy	19120492
2	Huỳnh Quốc Duy	19120494
3	Phạm Đức Huy	19120534
4	Lê Thành Lộc	19120562

2. Bảng phân công công việc:

MSSV	Họ tên	Công việc
19120492	Đỗ Thái Duy	Chụp hình từng bước quá trình cài đặt, mô tả và chú thích quá trình thực hiện vào báo cáo.
19120494	Huỳnh Quốc Duy	Thực hiện chạy thêm các ví dụ và bộ dữ liệu khác trên các thuật toán đã triển khai.
19120534	Phạm Đức Huy	Thực hiện chạy thêm các ví dụ và bộ dữ liệu khác trên các thuật toán đã triển khai.
19120562	Lê Thành Lộc	Chụp hình từng bước quá trình cài đặt, mô tả và chú thích quá trình thực hiện vào báo cáo.

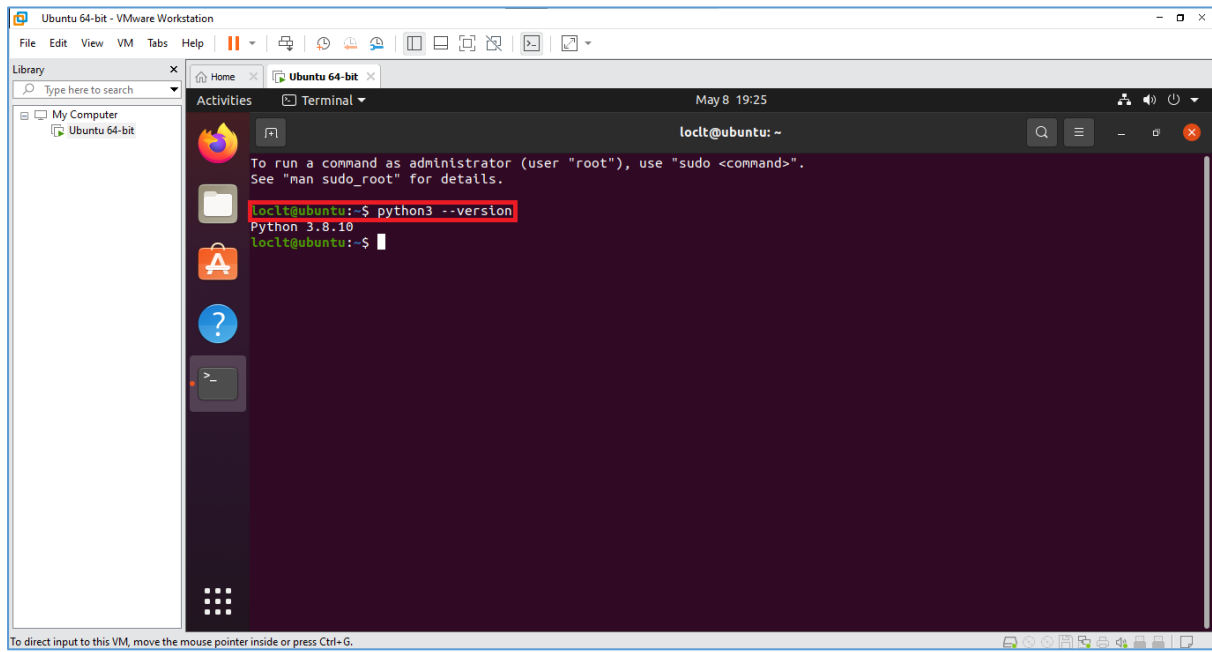
3. Đánh giá mức độ hoàn thành:

STT	Họ tên	MSSV	Mức độ hoàn thành
1	Đỗ Thái Duy	19120492	100%
2	Huỳnh Quốc Duy	19120494	100%
3	Phạm Đức Huy	19120534	100%
4	Lê Thành Lộc	19120562	100%

II. Nội dung tìm hiểu

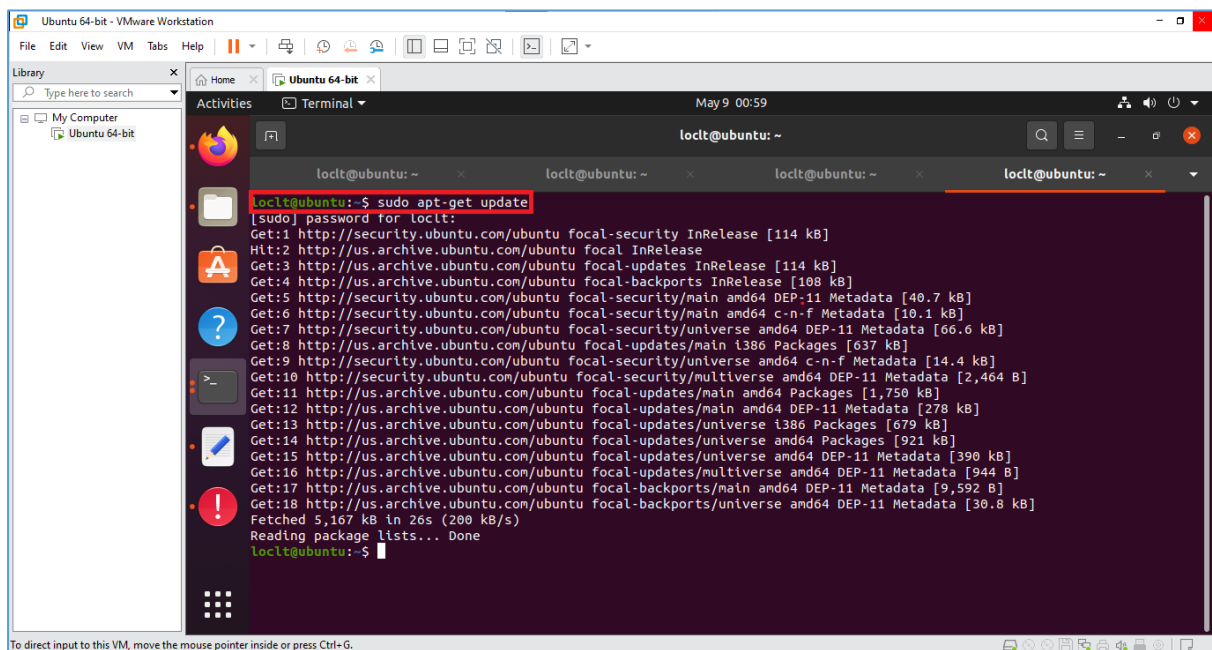
1. Cài đặt Spark trên Linux:

Bước 1: Kiểm tra xem đã cài python chưa.



```
loc@ubuntu: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
loc@ubuntu:~$ python3 --version  
Python 3.8.10  
loc@ubuntu:~$
```

Bước 2: Cập nhật các chỉ mục gói của hệ điều hành Linux.

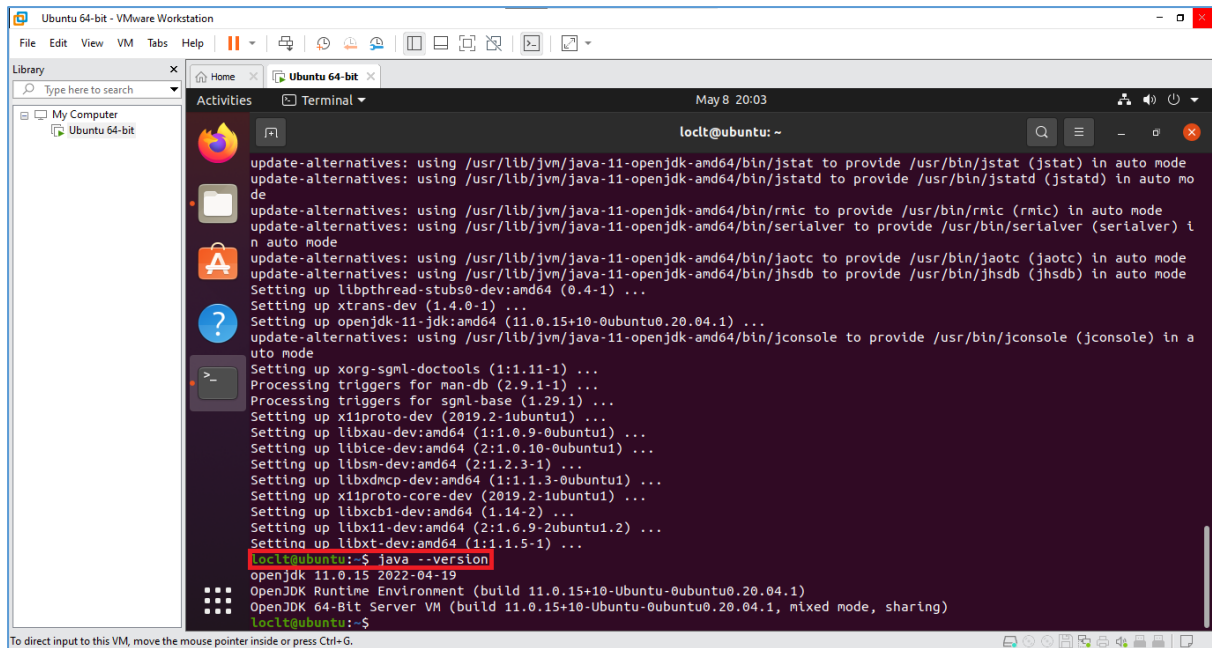


```
loc@ubuntu: ~  
loc@ubuntu: ~  
loc@ubuntu: ~  
loc@ubuntu: ~  
loc@ubuntu:~$ sudo apt-get update  
[sudo] password for loc:  
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]  
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease  
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]  
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [40.7 kB]  
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [10.1 kB]  
Get:7 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [66.6 kB]  
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [637 kB]  
Get:9 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [14.4 kB]  
Get:10 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 B]  
Get:11 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,750 kB]  
Get:12 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [278 kB]  
Get:13 http://us.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [679 kB]  
Get:14 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [921 kB]  
Get:15 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [390 kB]  
Get:16 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]  
Get:17 http://us.archive.ubuntu.com/ubuntu focal-backports/main amd64 DEP-11 Metadata [9,592 B]  
Get:18 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [30.8 kB]  
Fetched 5,167 kB in 26s (200 kB/s)  
Reading package lists... Done  
loc@ubuntu:~$
```

Bước 3: Cài đặt Java Runtime Environment (JRE) và Java Development Kit (JDK), sau đó kiểm tra xem phiên bản Java được cài đặt có đủ tiêu chuẩn cài spark không (Java phải từ phiên bản 8 trở lên).

```
loclt@ubuntu:~$ sudo apt-get install openjdk-11-jre
[sudo] password for loclt:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni openjdk-11-jre-headless
Suggested packages:
  default-jre fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni openjdk-11-jre openjdk-11-jre-headless
0 upgraded, 7 newly installed, 0 to remove and 113 not upgraded.
Need to get 39.6 MB of archives.
After this operation, 179 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 java-common all 0.72 [6,816 B]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 openjdk-11-jre-headless amd64 11.0.15+10-0ubuntu0.20.04.1 [37.3 MB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal/main amd64 ca-certificates-java all 20190405ubuntu1 [12.2 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal/main amd64 fonts-dejavu-extra all 2.37-1 [1,953 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libatk-wrapper-java all 0.37.1-1 [53.0 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libatk-wrapper-java-jni amd64 0.37.1-1 [45.1 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 openjdk-11-jre amd64 11.0.15+10-0ubuntu0.20.04.1 [17.5 kB]
Fetched 39.6 MB in 3min 16s (202 kB/s)
Selecting previously unselected package java-common.
(Reading database ... 155644 files and directories currently installed.)
```

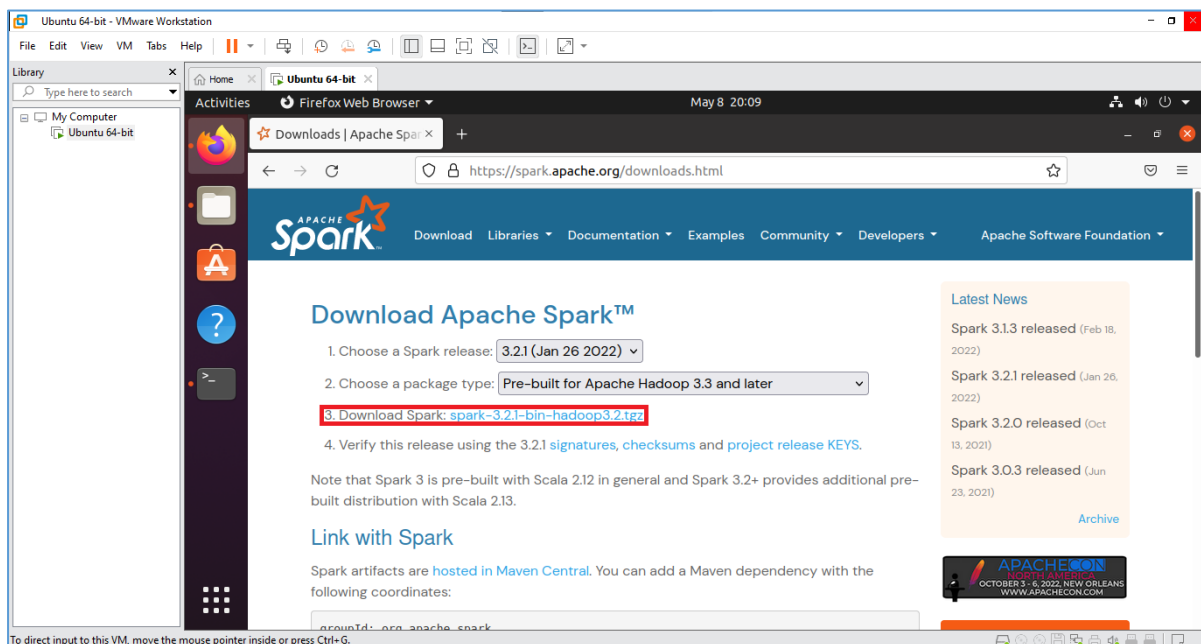
```
loclt@ubuntu:~$ sudo apt-get install openjdk-11-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev
  openjdk-11-jdk-headless x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-11-demo openjdk-11-source visualvm
The following NEW packages will be installed:
  libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-11-jdk
  openjdk-11-jdk-headless x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 14 newly installed, 0 to remove and 113 not upgraded.
Need to get 228 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 xorg-sgml-doctools all 1:1.11-1 [12.9 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/main amd64 x11proto-dev all 2019.2-1ubuntu1 [594 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal/main amd64 x11proto-core-dev all 2019.2-1ubuntu1 [2,620 B]
Get:4 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libice-dev amd64 2:1.0.10-0ubuntu1 [47.8 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libpthread-stubs0-dev amd64 0.4-1 [5,384 B]
Get:6 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libsm-dev amd64 2:1.2.3-1 [17.0 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libxau-dev amd64 1:1.0.9-0ubuntu1 [9,552 B]
Get:8 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libxdmcp-dev amd64 1:1.1.3-0ubuntu1 [25.3 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal/main amd64 xtrans-dev all 1.4.0-1 [68.9 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libxcb1-dev amd64 1.14-2 [80.5 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libx11-dev amd64 2:1.6.9-2ubuntu1.2 [647 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libxt-dev amd64 1:1.1.5-1 [395 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 openjdk-11-jdk-headless amd64 11.0.15+10-0ubuntu0.20.04.1 [223 MB]
Get:14 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 openjdk-11-jdk amd64 11.0.15+10-0ubuntu0.20.04.1 [223 MB]
```

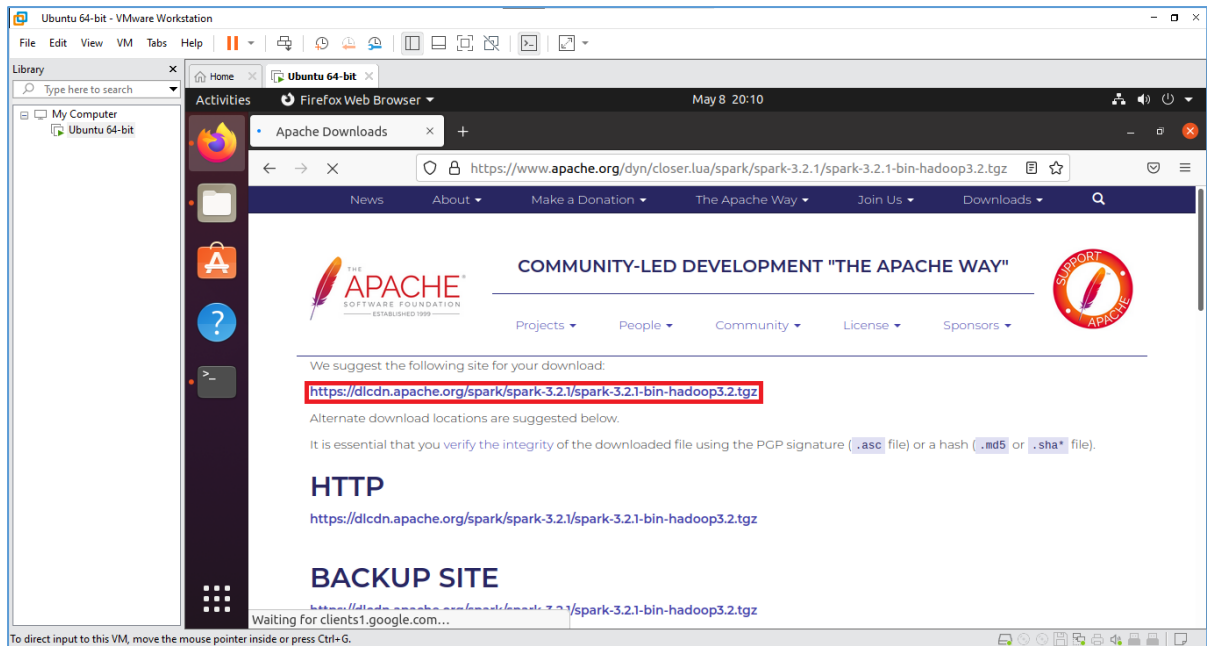


Bước 4: Tải và cài đặt spark từ link trên web.

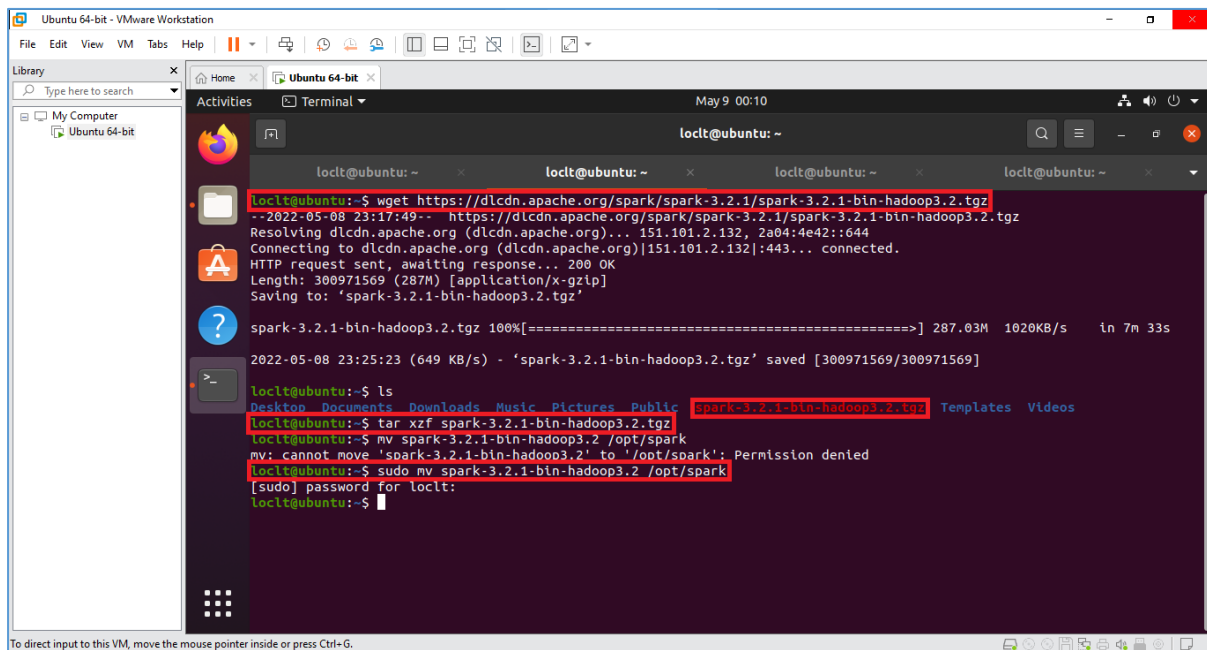
- Truy cập trang web <https://spark.apache.org/downloads.html> và click vào đường dẫn ở mục **3.Download spark** để lấy link tải spark:

<https://d1cdn.apache.org/spark/spark-3.2.1/spark-3.2.1-bin-hadoop3.2.tgz>



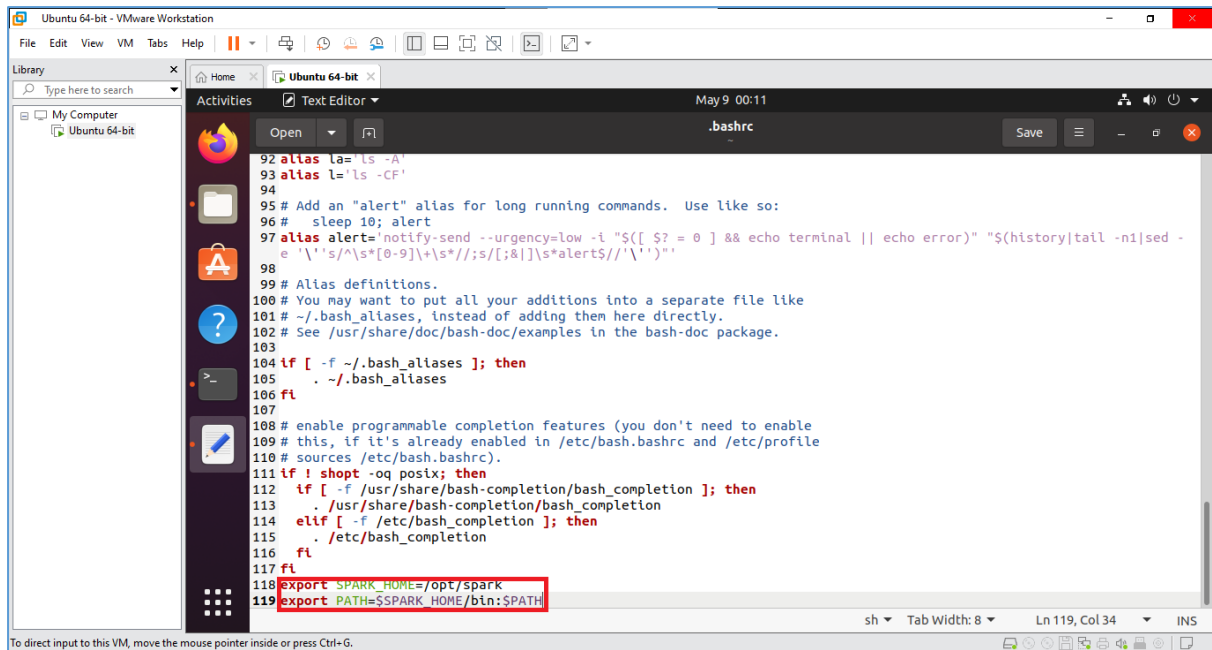


- Cài đặt spark từ link vừa lấy được ở trên và tiến hành giải nén với lệnh `tar xzf`, sau đó chuyển thư mục giải nén vào thư mục `/opt`.



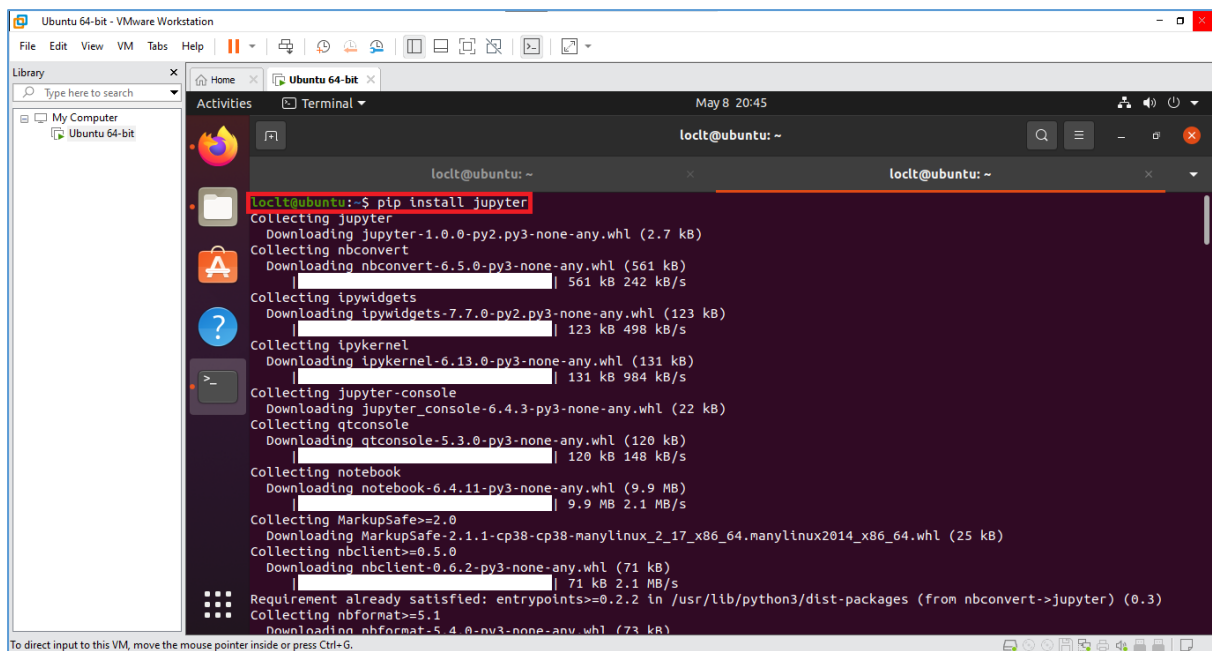
Bước 5: Thực hiện chỉnh sửa file `.bashrc`

- Nhập lệnh `gedit ~/.bashrc` để mở file `.bashrc`.
- Thêm các dòng lệnh export như hình vào cuối file.
- **Ctrl + S** để lưu thay đổi.

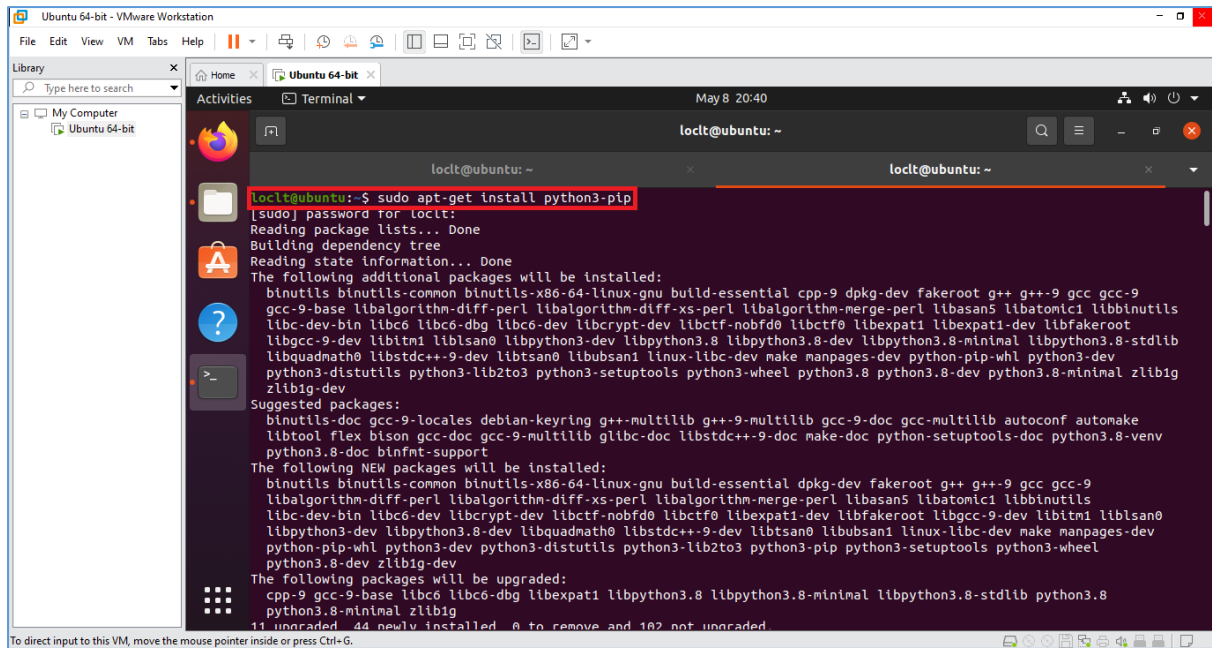


```
92 alias la='ls -A'
93 alias l='ls -CF'
94
95 # Add an "alert" alias for long running commands. Use like so:
96 # sleep 10; alert
97 alias alert='notify-send --urgency=low -i "${?} = 0" && echo terminal || echo error" "${history|tail -n1|sed -
98 e '\''s/^s*[0-9]+\s*//;s/[:&]]\s*alert$//'\''"'
99 # Alias definitions.
100 # You may want to put all your additions into a separate file like
101 # ~/.bash_aliases, instead of adding them here directly.
102 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
103
104 if [ -f ~/.bash_aliases ]; then
105     . ~/.bash_aliases
106 fi
107
108 # enable programmable completion features (you don't need to enable
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112     if [ -f /usr/share/bash-completion/bash_completion ]; then
113         . /usr/share/bash-completion/bash_completion
114     elif [ -f /etc/bash_completion ]; then
115         . /etc/bash_completion
116     fi
117 fi
118 export SPARK_HOME=/opt/spark
119 export PATH=$SPARK_HOME/bin:$PATH
```

Bước 6: Tiến hành tải **pip** và tải **jupyter** với **pip**.

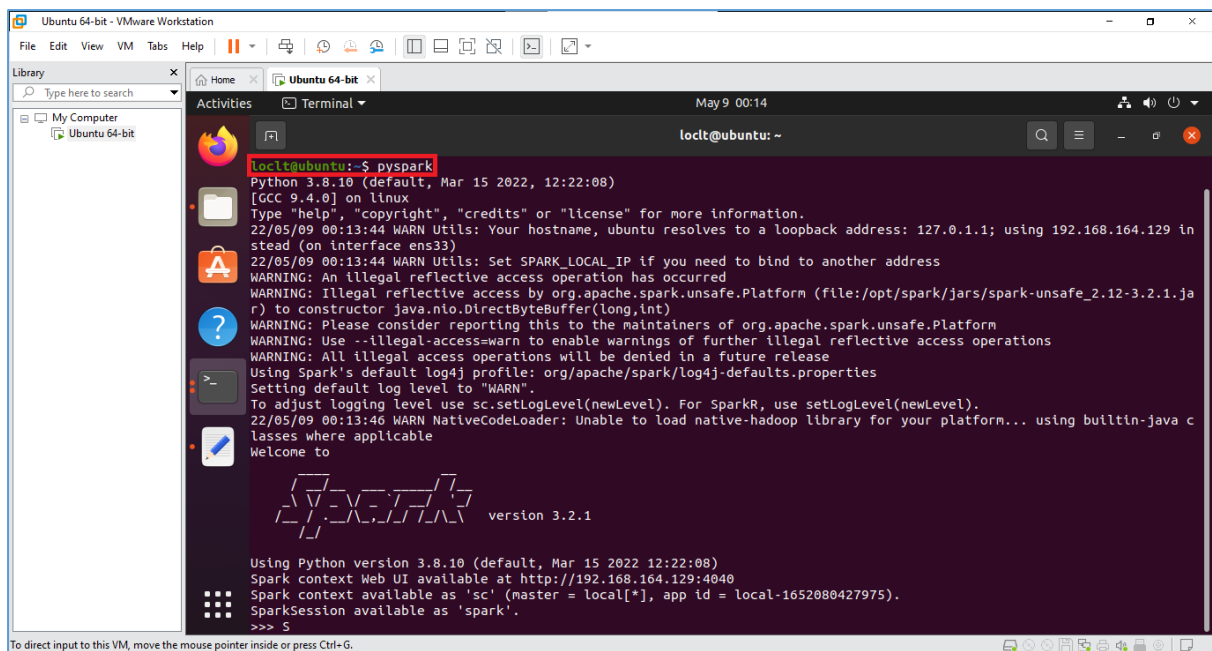


```
locit@ubuntu: ~
locit@ubuntu: ~
locit@ubuntu: ~
locit@ubuntu: ~$ pip install jupyter
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting nbconvert
  Downloading nbconvert-6.5.0-py3-none-any.whl (561 kB)
Collecting ipywidgets
  Downloading ipywidgets-7.7.0-py2.py3-none-any.whl (123 kB)
Collecting ipykernel
  Downloading ipykernel-6.13.0-py3-none-any.whl (131 kB)
Collecting jupyter-console
  Downloading jupyter_console-6.4.3-py3-none-any.whl (22 kB)
Collecting qtconsole
  Downloading qtconsole-5.3.0-py3-none-any.whl (120 kB)
Collecting notebook
  Downloading notebook-6.4.11-py3-none-any.whl (9.9 MB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.1-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Collecting nbclient>=0.5.0
  Downloading nbclient-0.6.2-py3-none-any.whl (71 kB)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/lib/python3/dist-packages (from nbconvert->jupyter) (0.3)
Collecting nbformat>=5.1
  Downloading nbformat-5.4.0-py3-none-any.whl (73 kB)
```

```
loclt@ubuntu: ~  
$ sudo apt-get install python3-pip  
[sudo] password for loclt:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9  
  gcc-9-base libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils  
  libc-dev-bin libc6 libc6-dbg libc6-dev libcrypt-dev libctf-nobfd0 libctf0 libexpat1 libexpat1-dev libfakeroot  
  libgcc-9-dev libitm1 liblsan0 libpython3-dev libpython3.8 libpython3.8-dev libpython3.8-minimal libpython3.8-stdlib  
  libquadmath0 libstdc++-9-dev libstdc++6 libubsan1 linux-libc-dev make manpages-dev python-pip-whl python3-dev  
  python3-distutils python3-lib2to3 python3-setuptools python3-wheel python3.8 python3.8-dev python3.8-minimal zlib1g  
  zlib1g-dev  
Suggested packages:  
  binutils-doc gcc-9-locales debian-keyring g++-multilib g++-9-multilib gcc-9-doc gcc-multilib autoconf automake  
  libtool flex bison gcc-doc gcc-9-multilib glibc-doc libstdc++-9-doc make-doc python-setuptools-doc python3.8-venv  
  python3.8-doc binfmt-support  
The following NEW packages will be installed:  
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-9 gcc gcc-9  
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils  
  libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0 libexpat1-dev libfakeroot libgcc-9-dev libitm1 liblsan0  
  libpython3-dev libpython3.8-dev libquadmath0 libstdc++-9-dev libstdc++6 libubsan1 linux-libc-dev make manpages-dev  
  python-pip-whl python3-dev python3-distutils python3-lib2to3 python3-pip python3-setuptools python3-wheel  
  python3.8-dev python3.8-minimal zlib1g-dev  
The following packages will be upgraded:  
  cpp-9 gcc-9-base libc6 libc6-dbg libexpat1 libpython3.8 libpython3.8-minimal libpython3.8-stdlib python3.8  
  python3.8-minimal zlib1g  
11 upgraded, 44 newly installed, 0 to remove and 102 not upgraded.
```

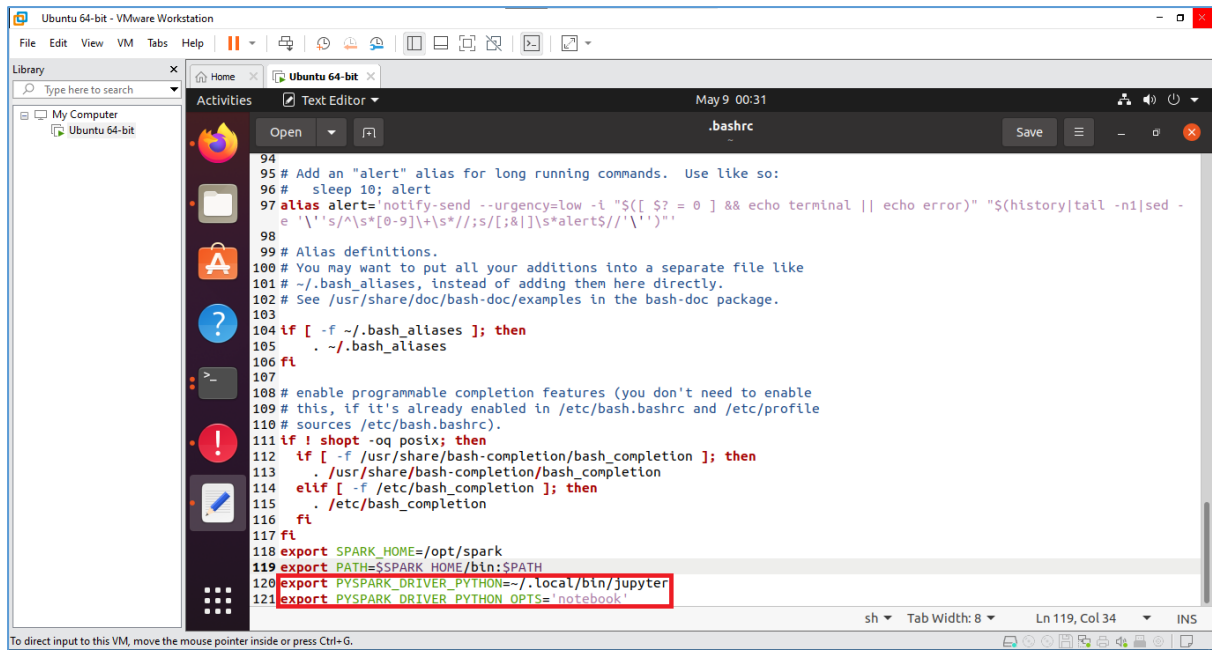
Bước 7: Khởi động và chạy pyspark.



```
loclt@ubuntu: ~  
$ pyspark  
Python 3.8.10 (default, Mar 15 2022, 12:22:08)  
[GCC 9.4.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
22/05/09 00:13:44 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.164.129 in  
stead (on interface ens33)  
22/05/09 00:13:44 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address  
WARNING: An illegal reflective access operation has occurred  
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/jars/spark-unsafe_2.12-3.2.1.jar  
to constructor java.nio.DirectByteBuffer(long,int)  
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform  
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations  
WARNING: All illegal access operations will be denied in a future release  
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).  
22/05/09 00:13:46 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java c  
lasses where applicable  
Welcome to  
  
Spark version 3.2.1  
  
Using Python version 3.8.10 (default, Mar 15 2022 12:22:08)  
Spark context Web UI available at http://192.168.164.129:4040  
Spark context available as 'sc' (master = local[*], app id = local-1652080427975).  
SparkSession available as 'spark'.  
>>> S
```

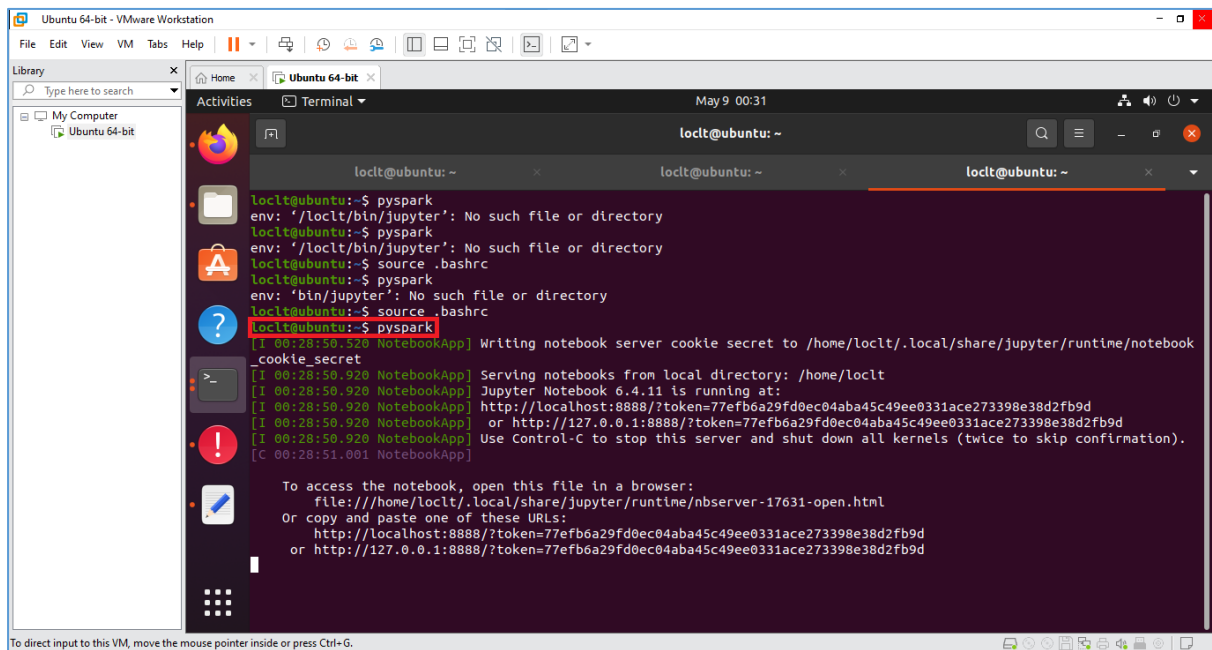
Bước 8: Tiếp tục chỉnh sửa file .bashrc để có thể sử dụng jupyter notebook

- Nhập lệnh `gedit ~/.bashrc` để mở file `.bashrc`.
- Thêm các dòng lệnh export như hình vào cuối file.
- **Ctrl + S** để lưu thay đổi.



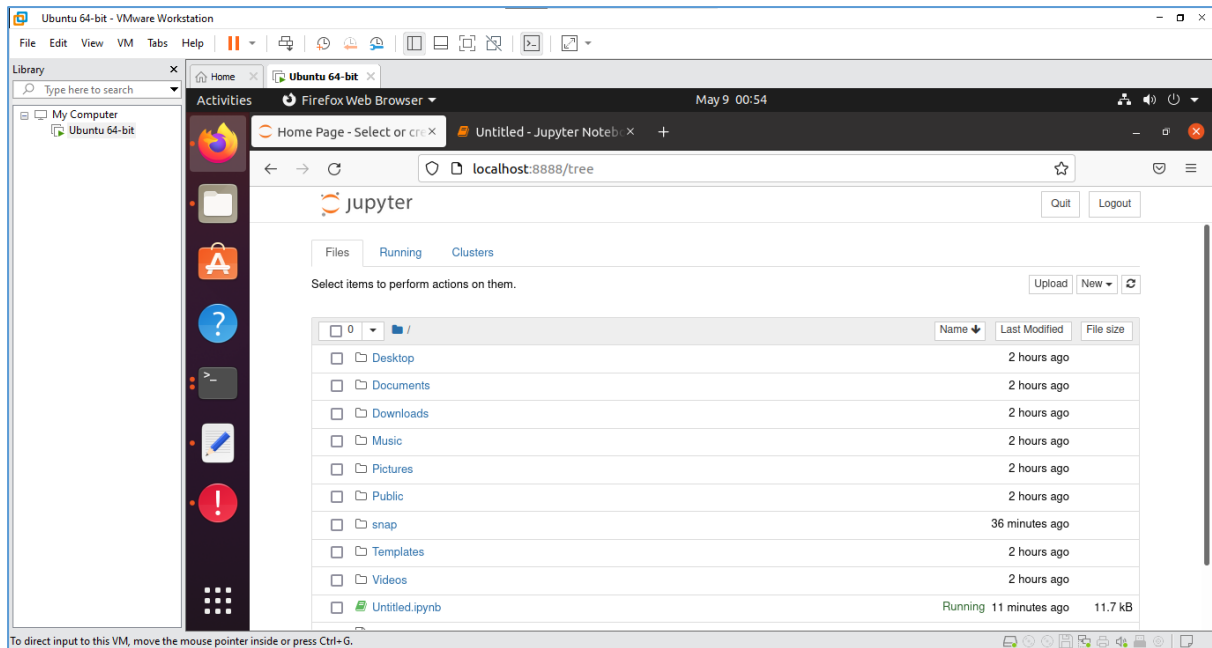
```
94
95 # Add an "alert" alias for long running commands. Use like so:
96 # sleep 10; alert
97 alias alert='notify-send --urgency=low -i "${[ $? = 0 ]} && echo terminal || echo error)" "$(history|tail -n1|sed -
98 e '\''s/^s*[0-9]\+s*//;s/[:&]\+s*alert$//'\''")'
99
100 # Alias definitions.
101 # You may want to put all your additions into a separate file like
102 # ~/.bash_aliases, instead of adding them here directly.
103 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
104
105 if [ -f ~/.bash_aliases ]; then
106 . ~/.bash_aliases
107 fi
108
109 # enable programmable completion features (you don't need to enable
110 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
111 # sources /etc/bash.bashrc).
112 if ! shopt -oq posix; then
113 if [ -f /usr/share/bash-completion/bash_completion ]; then
114 . /usr/share/bash-completion/bash_completion
115 elif [ -f /etc/bash_completion ]; then
116 . /etc/bash_completion
117 fi
118 fi
119 export SPARK_HOME=/opt/spark
120 export PATH=$SPARK_HOME/bin:$PATH
121 export PYSPARK_DRIVER_PYTHON=./local/bin/jupyter
122 export PYSPARK_DRIVER_PYTHON_OPTS='notebook'
```

Bước 9: Nhập lệnh *pyspark* để mở **jupyter notebook** và thực hiện chạy spark trên **jupyter notebook**.



```
loclt@ubuntu:~$ pyspark
env: '/loclt/bin/jupyter': No such file or directory
loclt@ubuntu:~$ pyspark
env: '/loclt/bin/jupyter': No such file or directory
loclt@ubuntu:~$ source .bashrc
loclt@ubuntu:~$ pyspark
env: 'bin/jupyter': No such file or directory
loclt@ubuntu:~$ source .bashrc
loclt@ubuntu:~$ pyspark
[I 00:28:50.520 NotebookApp] Writing notebook server cookie secret to /home/loclt/.local/share/jupyter/runtime/notebook
_cookie_secret
[I 00:28:50.920 NotebookApp] Serving notebooks from local directory: /home/loclt
[I 00:28:50.920 NotebookApp] 0.0.1:8888/?token=77efb6a29fd0ec04aba45c49ee0331ace273398e38d2fb9d
[I 00:28:50.920 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

To access the notebook, open this file in a browser:
file:///home/loclt/.local/share/jupyter/runtime/nbserver-17631-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=77efb6a29fd0ec04aba45c49ee0331ace273398e38d2fb9d
or http://127.0.0.1:8888/?token=77efb6a29fd0ec04aba45c49ee0331ace273398e38d2fb9d
```



2. Cài đặt thuật toán và thực hiện chạy các ví dụ và bộ dữ liệu khác:

- a) Các giải thuật học máy trên bộ dữ liệu **mushroom.csv** được public trên trang kaggle:

Mô tả: Bài toán phân lớp các loại nấm dựa vào đặc điểm của chúng. Dữ liệu gồm 23 cột. Trong bài làm này để đơn giản. Ta sẽ lấy 10 cột dữ liệu và 1 cột đích để phân lớp bài toán

Các bước thực hiện:

Bước 1: Đọc dữ liệu lên từ file **mushroom.csv**.

```
data = spark.read.load("/kaggle/input/mushroom-classification/mushrooms.csv", format="csv", header=True,
data = data.withColumn("cap-shape", data["cap-shape"]).withColumn("label", data["class"]). \
    withColumn("cap-surface", data["cap-surface"]). \
    withColumn("cap-color", data["cap-color"]). \
    withColumn("bruises", data["bruises"]). \
    withColumn("odor", data["odor"]). \
    withColumn("gill-attachment", data["gill-attachment"]). \
    withColumn("gill-spacing", data["gill-spacing"]). \
    withColumn("gill-size", data["gill-size"]). \
    withColumn("gill-color", data["gill-color"]). \
    withColumn("stalk-shape", data["stalk-shape"])
# withColumn("BMI", data["Body mass index"] - 0)
```

Bước 2: Sử dụng **onehot encoding** để mã hóa các cột dữ liệu từ dạng chuỗi về dạng số.

```
# one hot encoding
from pyspark.ml.feature import VectorAssembler, StringIndexer, OneHotEncoder

stringIndexer = StringIndexer().setInputCol("cap-shape").setOutputCol("cap-shape2")
ageModel = stringIndexer.fit(data)
data = ageModel.transform(data)
stringIndexer = StringIndexer().setInputCol("cap-surface").setOutputCol("cap-surface2")
ageModel = stringIndexer.fit(data)
data = ageModel.transform(data)
stringIndexer = StringIndexer().setInputCol("cap-color").setOutputCol("cap-color2")
ageModel = stringIndexer.fit(data)
data = ageModel.transform(data)
stringIndexer = StringIndexer().setInputCol("bruises").setOutputCol("bruises2")
ageModel = stringIndexer.fit(data)
data = ageModel.transform(data)
stringIndexer = StringIndexer().setInputCol("odor").setOutputCol("odor2")
ageModel = stringIndexer.fit(data)
data = ageModel.transform(data)
stringIndexer = StringIndexer().setInputCol("gill-attachment").setOutputCol("gill-attachment2")
ageModel = stringIndexer.fit(data)
data = ageModel.transform(data)
stringIndexer = StringIndexer().setInputCol("gill-spacing").setOutputCol("gill-spacing2")
ageModel = stringIndexer.fit(data)
data = ageModel.transform(data)
```

Bước 3: Tập hợp mảng vector các input gồm 10 cột dữ liệu vừa được mã hóa.

```
assem = VectorAssembler(inputCols=["cap-shape2", "cap-surface2", "cap-color2", "bruises2", "odor2", "gill-attachment2", "gill-spacing2", "gill-size2", "gill-color2"])
data = assem.transform(data)
# data.show()
```

Bước 4: Mã hóa cột đích 'label' thành cột 'indexedLabel' cũng như phân biệt các dữ liệu dạng categorical trong mảng vector input. Nếu một cột có hơn 4 giá trị khác nhau sẽ được xem là dữ liệu loại liên tục. Sau đó tách dữ liệu theo tỉ lệ 7:3.

```
# Fit on whole dataset to include all labels in index.
labelIndexer = StringIndexer(inputCol="label", outputCol="indexedLabel").fit(data)
# Automatically identify categorical features, and index them.
# We specify maxCategories so features with > 4 distinct values are treated as continuous.
featureIndexer =\
    VectorIndexer(inputCol="features", outputCol="indexedFeatures", maxCategories=4).fit(data)
# data.show()
# # Split the data into training and test sets (30% held out for testing)
(trainingData, testData) = data.randomSplit([0.7, 0.3])
```

Bước 5: Kết quả chạy trên các giải thuật

- *Decision tree:*

+ **Model:**

```
# # Train a DecisionTree model.
dt = DecisionTreeClassifier(labelCol="indexedLabel", featuresCol="indexedFeatures")
```

+ **Kết quả:** Độ chính xác của giải thuật: 0.984755%.

```
+-----+-----+
|prediction|indexedLabel|      features|
+-----+-----+
|      0.0|          0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|          0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|          0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|          0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|          0.0|[3.0,2.0,1.0,0.0,...|
+-----+-----+
```

only showing top 5 rows

```
DecisionTreeClassificationModel: uid=DecisionTreeClassifier_f3974bb1c45b, depth=5, numNodes=11, numClasses=2, numFeatures=10
Decision Tree - Test Accuracy = 0.984755
Decision Tree - Test Error = 0.0152452
The Confusion Matrix for Decision Tree Model is :
[[1251  0]
 [ 37 1139]]
The precision score for Decision Tree Model is: 0.984754841367944
The recall score for Decision Tree Model is: 0.984754841367944
```

- **Naive Bayes**

+ **Model:**

```
# create the trainer and set its parameters
nb = NaiveBayes(labelCol="label12", featuresCol="features", smoothing=1.0, modelType="multinomial")
```

+ **Kết quả:** Độ chính xác của giải thuật: 0.742611%.

```
+-----+-----+
|prediction|label12|      features|
+-----+-----+
|      0.0|      0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|      0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|      0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|      0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|      0.0|[3.0,2.0,1.0,0.0,...|
+-----+-----+
```

only showing top 5 rows

```
Naive Bayes - Test set accuracy = 0.7426108374384236
The Confusion Matrix for Naive Bayes Model is :
[[996 265]
 [362 813]]
The precision score for Naive Bayes Model is: 0.7426108374384236
The recall score for Naive Bayes Model is: 0.7426108374384236
```

- **RandomForest**

+ **Model:**

```
# create the trainer and set its parameters
rf = RandomForestClassifier(labelCol="label12", featuresCol="features", numTrees=10)
```

+ **Kết quả:** Độ chính xác của giải thuật: 0.988333%.

```
+-----+-----+-----+
|prediction|label2|      features|
+-----+-----+-----+
|      0.0|      0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|      0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|      0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|      0.0|[3.0,2.0,1.0,0.0,...|
|      0.0|      0.0|[3.0,2.0,1.0,0.0,...|
+-----+-----+-----+
only showing top 5 rows
```

```
Naive Bayes - Test set accuracy = 0.9883333333333333
The Confusion Matrix for Naive Bayes Model is :
[[1242   0]
 [  28 1130]]
The precision score for Naive Bayes Model is: 0.9883333333333333
The recall score for Naive Bayes Model is: 0.9883333333333333
```

→ Đây là giải thuật có độ chính xác cao nhất trong bài toán này.

b) Ảnh cho các giải thuật trên các giải thuật học máy trên bộ dữ liệu **iris.csv** được public trên trang Kaggle:

Mô tả: Bài toán phân biệt loại hoa diên vĩ dựa vào các đặc điểm của chúng. Dữ liệu gồm 5 cột. Trong bài làm này, ta sẽ lấy 3 cột dữ liệu và 1 cột đích (species) để phân lớp bài toán

Các bước thực hiện:

Bước 1: Đọc dữ liệu từ file **iris.csv**. Sau đó đưa 3 cột dữ liệu đầu tiên từ dạng chuỗi về dạng số.

```
data = spark.read.load("iris.csv", format="csv", header=True, delimiter=",")
data = data.withColumn("sepal_length", data["sepal_length"] - 0). \
    withColumn("sepal_width", data["sepal_width"] - 0). \
    withColumn("petal_length", data["petal_length"] - 0)
```

Bước 2: Tập hợp mảng vector các input gồm 3 cột dữ liệu ở dạng số. Ta cũng mã hóa biến species bằng *StringIndexer*, tên của mỗi loại hoa sẽ được đánh số.

```
assem = VectorAssembler(inputCols=["sepal_length", "sepal_width", "petal_length"], outputCol='features')
data = assem.transform(data)

# Index Labels, adding metadata to the label column.
# Fit on whole dataset to include all labels in index.
labelIndexer = StringIndexer(inputCol="species", outputCol="indexedLabel").fit(data)
```

Bước 3: Đến đây đối với mỗi thuật toán ta sẽ biến đổi khác nhau:

- **DECISION TREE**

Bước 3.1: Tiếp theo ta dùng *VectorIndexer* để tìm các biến categorical trong vector input (nếu cột dữ liệu có ít hơn 4 giá trị khác nhau thì sẽ là categorical). Sau

đó đánh số các vector này. Ta sẽ chia dữ liệu theo tỉ lệ 7:3.

```
# We specify maxCategories so features with > 4 distinct values are treated as continuous.
featureIndexer = \
    VectorIndexer(inputCol="features", outputCol="indexedFeatures", maxCategories=4).fit(data)

# Split the data into training and test sets (30% held out for testing)
(trainingData, testData) = data.randomSplit([0.7, 0.3])
```

Bước 3.2: Tạo 1 mô hình **Decision Tree**. Sau đó dùng các phép biến đổi dữ liệu lên training data. Cuối cùng ta thu được *predictions* là các giá trị dự đoán được bằng thuật toán cây quyết định,

```
# Train a DecisionTree model.
dt = DecisionTreeClassifier(labelCol="indexedLabel", featuresCol="indexedFeatures")

# Chain indexers and tree in a Pipeline
pipeline = Pipeline(stages=[labelIndexer, featureIndexer, dt])

# Train model. This also runs the indexers.
model = pipeline.fit(trainingData)

# Make predictions.
predictions = model.transform(testData)
```

+ **Kết quả:** Độ chính xác của thuật toán: **93.75%**.

```
DecisionTreeClassificationModel: uid=DecisionTreeClassifier_d2fc15112d0d, depth=5, numNodes=13, numClasses=3, numFeatures=3
Decision Tree - Test Accuracy = 0.9375
Decision Tree - Test Error = 0.0625
The Confusion Matrix for Decision Tree Model is :
[[13  0  0]
 [ 0 20  1]
 [ 0  2 12]]
The precision score for Decision Tree Model is: 0.9375
The recall score for Decision Tree Model is: 0.9375
```

- **NAIVE BAYES**

Bước 3.1: Sau khi dùng StringIndexer để biến đổi biến đích species thì ta cũng chia dữ liệu rồi tạo một mô hình Naive Bayes rồi áp dụng các phép biến đổi dữ liệu.


```
splits = data.randomSplit([0.7, 0.3], 1000)
train = splits[0]
test = splits[1]

# create the trainer and set its parameters
nb = NaiveBayes(smoothing=1.0, modelType="multinomial")

# train the model
#model = nb.fit(train)

# Chain indexers and tree in a Pipeline
pipeline = Pipeline(stages=[labelIndexer, nb])

# Train model. This also runs the indexers.
model = pipeline.fit(train)

# select example rows to display.
predictions = model.transform(test)
```

+ **Kết quả:** Độ chính xác của thuật toán này là **90.7%**.

```
Naive Bayes - Test set accuracy = 0.9069767441860465
The Confusion Matrix for Naive Bayes Model is :
[[14  0  0]
 [ 0 11  4]
 [ 0  0 14]]
The precision score for Naive Bayes Model is: 0.9069767441860465
The recall score for Naive Bayes Model is: 0.9069767441860465
```

- **RANDOM FOREST**

Bước 3.1: RandomForest cũng thực hiện các biến tương tự như **DecisionTree**.

Chỉ khác là ở đây ta tạo một mô hình **RandomForestClassifier**.

```
# We specify maxCategories so features with 2 distinct values are treated as continuous
featureIndexer = \
    VectorIndexer(inputCol="features", outputCol="indexedFeatures", maxCategories=4).fit(data)

# Split the data into training and test sets (30% held out for testing)
(trainingData, testData) = data.randomSplit([0.7, 0.3])

# Train a RandomForest model.
rf = RandomForestClassifier(labelCol="indexedLabel", featuresCol="indexedFeatures", numTrees=10)

# Chain indexers and forest in a Pipeline
pipeline = Pipeline(stages=[labelIndexer, featureIndexer, rf])

# Train model. This also runs the indexers.
model = pipeline.fit(trainingData)

# Make predictions.
predictions = model.transform(testData)
```


+ **Kết quả:** Độ chính xác của thuật toán này là **96.2%**.

```
RandomForestClassificationModel: uid=RandomForestClassifier_0380fe8fe874, numTrees=10, numClasses=3, numFeatures=3
Random Forest - Test Accuracy = 0.962264
Random Forest - Test Error = 0.0377358
The Confusion Matrix for Random Forest Model is :
[[27  0  0]
 [ 0 10  1]
 [ 0  1 14]]
The precision score for Random Forest Model is: 0.9622641509433962
The recall score for Random Forest Model is: 0.9622641509433962
```

→ **Đối với bài toán này, RandomForest cũng là thuật toán có độ chính xác cao nhất.**

III. Tài liệu tham khảo

- [1] https://github.com/Ruthvicp/CS5590_BigDataProgramming/wiki/Lab-Assignment-4----Spark-MLlib-classification-algorithms,-word-count-on-twitter-streaming
- [2] https://changhsinlee.com/install-pyspark-windows-jupyter/?fbclid=IwAR1o42fIl98kPc_28NyBFtMerWOIPLZm5gFevHll8cCKJOVyvqHg20xcCwY
- [3] <https://www.sicara.fr/blog/2017-05-02-get-started-pyspark-jupyter-notebook-3-minutes>