خطوات تثبيت ال xgboost

من المهم جدا تعطل أي برنامج ال antivirus او anti-malware

In order to install and use XGBoost with Python you need three software on your windows machine:

- A Python installation Anaconda.
- Git Git for Windows.
- MINGW MinGW-W64
-

I assume you have Anaconda up and running. I am using Anaconda for Python 3.6.x

Git installation is quite easy. There are several options, one is to use Git for Windows. Just download and save the installer file on your disk, then launch it by double clicking it. Then follow the installer instructions.

Once the installation has completed look for a program called **Git Bash** in your start menu. Launch it. It starts a terminal running the **Bash shell (you need to know basic Unix commands)**

This is different from the regular Windows terminal, but it is more handy for what we need to do.

First, go to the directory where you want to save XGBoost code by typing the `cd` command in the **bash terminal**. I used the following.

```
      $ cd /c/Users/tamer/
```

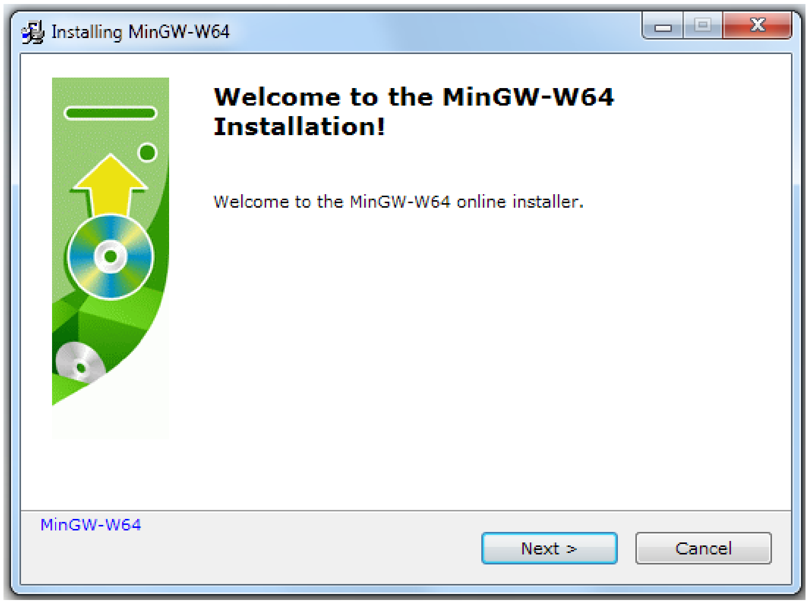
Then download XGBoost by typing the following commands.

```
$ git clone --recursive https://github.com/dmlc/xgboost
$ cd xgboost
$ git submodule init
$ git submodule update
```

Next step is to build XGBoost on your machine, i.e. compile the code we just downloaded.

For this we need a compiler provided with MinGW-W64. I downloaded the installer from this link.

Save the file on your disk, then launch it by double clicking it. You may need to authorize this operation. Then click next on the first screen:

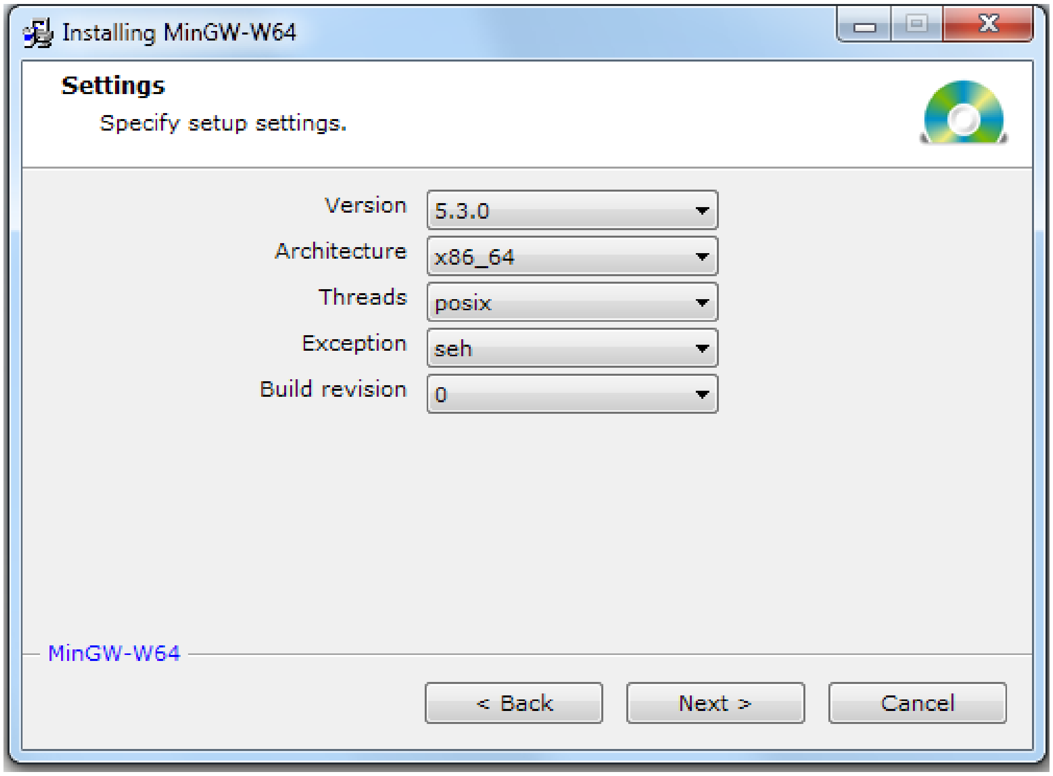


Installing MinGW-W64

Welcome to the MinGW-W64 Installation!

Welcome to the MinGW-W64 online installer.

Then select the x86_64 item in the architecture menu. Do not modify the other settings.



Installing MinGW-W64

Settings

Specify setup settings.

Version: 5.3.0

Architecture: x86_64

Threads: posix

Exception: seh

Build revision: 0

Then click Next and follow the instructions. On my machine, it installed the compiler in the:

`C:\Program Files\mingw-w64\x86_64-5.3.0-posix-seh-rt_v4-rev0` directory.

The make command and the runtime libraries are in this directory (look for the directory that contains mingw32-make):

```
C:\Program Files\mingw-w64\x86_64-5.3.0-posix-seh-rt_v4-rev0\mingw64\bin
```

Add the above path to the Path system variable (Follow my video)

Then close the Git Bash terminal, and launch it again. This will consider the new Path variable. To check you are fine, type the following:

In **bash terminal**

```
$ which mingw32-make
```

It should return something like:

```
/c/Program Files/mingw-w64/x86_64-5.3.0-posix-seh-rt_v4-rev0/mingw64/bin/mingw32-make
```

To make our life easier, let us alias it as follows:

```
$ alias make='mingw32-make'
```

We can now build XGBoost. We first go back to the directory where we downloaded it:

```
$ cd /c/Users/tamer/xgboost
```

The command given in the instructions does not work as I write this blog entry. Until this is fixed, we need to compile each sub module explicitly with the following commands. Wait until each make command is completed before typing the next command.

```
    $ cd dmlc-core
    $ make -j4
    $ cd ../rabit
    $ make lib/librabit_empty.a -j4
    $ cd ..
    $ cp make/mingw64.mk config.mk
    $ make -j4
```

Once the last command completes the build is done. We can now install the Python module. What follows depends on the Python distribution you are using. For Anaconda, I will simply use the Anaconda prompt, and type the following in it (after the prompt, in my case):

```
[Anaconda3] C:\Users\tamer>):
[Anaconda3] C:\Users\tamer>cd xgboost\python-package
```

The point is to move to the python-package directory of XGBoost. Then type:

```
[Anaconda3] C:\Users\tamer\xgboost\python-package>python setup.py install
```

افتح في ال spyder ملف فاضي و اكتب فيه البرنامج ده

اتأكد من الpath مكتوب مظبوط

Run the program (Ctrl + a .. Ctrl + enter)

انا كتبت بخط صغير علشان لازم السطر الثاني يبقى في نفس السطر

```
import os

mingw_path = 'C:\\Program Files\\mingw-w64\\x86_64-5.3.0-posix-seh-rt_v4-rev0\\mingw64\\bin'

os.environ['PATH'] = mingw_path + ';' + os.environ['PATH']
```

We can then import xgboost and run a small example.

```
import xgboost as gab
```

THE End 😉