C:\crewaiworkspace>pip install chroma-hnswlib

Collecting chroma-hnswlib

Using cached chroma\_hnswlib-0.7.6.tar.gz (32 kB)

Installing build dependencies ... done

Getting requirements to build wheel ... done

Preparing metadata (pyproject.toml) ... done

Requirement already satisfied: numpy in c:\users\furqu\anaconda3\lib\site-packages (from chroma-hnswlib) (1.26.4)

Building wheels for collected packages: chroma-hnswlib

Building wheel for chroma-hnswlib (pyproject.toml) ... error

error: subprocess-exited-with-error

× Building wheel for chroma-hnswlib (pyproject.toml) did not run successfully.

│ exit code: 1

╰─> [5 lines of output]

running bdist\_wheel

running build

running build\_ext

building 'hnswlib' extension

error: Microsoft Visual C++ 14.0 or greater is required. Get it with "Microsoft C++ Build Tools": https://visualstudio.microsoft.com/visual-cpp-build-tools/

[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.

ERROR: Failed building wheel for chroma-hnswlib

Failed to build chroma-hnswlib

ERROR: ERROR: Failed to build installable wheels for some pyproject.toml based projects (chroma-hnswlib)

C:\crewaiworkspace>pip install chroma-hnswlib

\* History restored

Microsoft Windows [Version 10.0.26100.3476]

(c) Microsoft Corporation. All rights reserved.

(base) C:\crewaiworkspace>conda activate venv/

(C:\crewaiworkspace\venv) C:\crewaiworkspace>pip install chroma-hnswlib

Collecting chroma-hnswlib

Using cached chroma\_hnswlib-0.7.6.tar.gz (32 kB)

Installing build dependencies ... done

Getting requirements to build wheel ... done

Preparing metadata (pyproject.toml) ... done

Collecting numpy (from chroma-hnswlib)

Using cached numpy-2.2.4-cp312-cp312-win\_amd64.whl.metadata (60 kB)

Using cached numpy-2.2.4-cp312-cp312-win\_amd64.whl (12.6 MB)

Building wheels for collected packages: chroma-hnswlib

Building wheel for chroma-hnswlib (pyproject.toml) ... error

error: subprocess-exited-with-error

× Building wheel for chroma-hnswlib (pyproject.toml) did not run successfully.

│ exit code: 1

╰─> [5 lines of output]

running bdist\_wheel

running build

running build\_ext

building 'hnswlib' extension

error: Microsoft Visual C++ 14.0 or greater is required. Get it with "Microsoft C++ Build Tools": https://visualstudio.microsoft.com/visual-cpp-build-tools/

[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.

ERROR: Failed building wheel for chroma-hnswlib

Failed to build chroma-hnswlib

ERROR: Failed to build installable wheels for some pyproject.toml based projects (chroma-hnswlib)

The error you’re encountering when trying to install chroma-hnswlib in your Python environment stems from a missing dependency required to build the package from source: **Microsoft Visual C++ 14.0 or greater**. This is a common issue on Windows when installing packages that include C++ extensions (like chroma-hnswlib, which relies on the HNSW library for efficient vector similarity search). The package’s build process needs a C++ compiler, and your system doesn’t have it installed.

Here’s a breakdown of the issue and how to resolve it, tailored to your setup (C:\crewaiworkspace\venv on Windows 10):

**Why This Happens**

* **chroma-hnswlib Requires Compilation**: The package isn’t available as a pre-built wheel for your specific Python version (likely Python 3.12, based on cp312 in the logs) and Windows architecture. When a pre-built wheel isn’t found, pip attempts to build it from source using pyproject.toml.
* **C++ Dependency**: Building the HNSW library (a C++ library wrapped in Python) requires a C++ compiler. On Windows, this typically means Microsoft Visual C++ Build Tools.
* **Error Message**: The log explicitly states:

text

CollapseWrapCopy

error: Microsoft Visual C++ 14.0 or greater is required. Get it with "Microsoft C++ Build Tools": https://visualstudio.microsoft.com/visual-cpp-build-tools/

This confirms the missing compiler is the root cause.

**Steps to Fix the Issue**

**1. Install Microsoft Visual C++ Build Tools**

* **Download**:
  + Go to [Microsoft Visual Studio Build Tools](https://visualstudio.microsoft.com/visual-cpp-build-tools/).
  + Click “Download Build Tools” and run the installer (vs\_buildtools.exe).
* **Install Required Components**:
  + In the installer, select the **“Desktop development with C++”** workload.
  + Ensure these are checked (minimum requirements):
    - MSVC v143 - VS 2022 C++ x64/x86 build tools (or latest version).
    - C++ CMake tools for Windows.
    - Windows 10/11 SDK (latest version).
  + Click “Install” (this might take ~5-10 GB of space and 10-20 minutes depending on your system).
* **Restart**: After installation, restart your computer to ensure the tools are fully integrated.

**2. Verify the Installation**

* Open a new Command Prompt and run:

bash

CollapseWrapCopy

cl

* If installed correctly, you’ll see output like:

text

CollapseWrapCopy

Microsoft (R) C/C++ Optimizing Compiler Version 19.xx.x.x for x86

* If you get “command not found,” the tools aren’t in your PATH—reinstall or check your environment variables.

**3. Retry the Installation**

* Activate your virtual environment:

bash

CollapseWrapCopy

C:\crewaiworkspace\venv\Scripts\activate

* Install chroma-hnswlib again:

bash

CollapseWrapCopy

pip install chroma-hnswlib

* With the C++ tools installed, the build should succeed, compiling the HNSW extension.

**Alternative Solutions**

If you’d rather avoid installing build tools or run into further issues, here are workarounds:

**1. Use a Pre-Built Wheel (If Available)**

* Check if a compatible wheel exists for your Python version and platform on PyPI:

bash

CollapseWrapCopy

pip install chroma-hnswlib --only-binary :all:

* This forces pip to use pre-built wheels only. However, as of chroma-hnswlib-0.7.6 (the version in your logs), wheels may not be available for Python 3.12 on Windows, triggering the source build.

**2. Downgrade Python**

* Python 3.12 is relatively new (released October 2023), and some packages lack pre-built wheels for it. Downgrading to Python 3.11 or 3.10 might find a compatible wheel:
  + Create a new virtual environment with Python 3.11: