CS 147 Final Project Report

David Strathman - 862170478
Team Buy The Dip

Project Overview

The goal of this project was to accelerate the process of statistical bootstrapping by using a GPU. Bootstrapping is the process where you randomly choose a subset of the data that you have to create a linear regression of the data. You repeat this process a significant amount of times, and based on the law of large numbers, this should reduce your error ranges and increase the accuracy of your confidence intervals.

I planned on using this concept on something that would take a significant amount of time to compute and be related to a topic that I thought would not be prevalent within a CS class. I chose to go with stock data (a lot of it) with daily adjusted closing prices of six tech companies since the year 2000 (NVIDIA being one of them :)). I would then regress this stock data against the DXY. The DXY is an index (a mathematical calculator based on several points of input data) that tracks the strength of the US dollar against other currencies around the globe. For example, a simple calculation would average the exchange rate between the USD and the British Pound and the USD against the Japanese Yen. (E.G. 100USD/1GBP and 50USD/1Yen = 75 DXY). When the value of the DXY index increases, this means that the US dollar is getting stronger, and when the DXY lowers in magnitude, this means that the US dollar is getting weaker.

The implications of this are that when the dollar is stronger, US based companies will find it cheaper to buy products from other countries that are denominated in another currency. For example, a factory in China that pays its workers in the Chinese Yuan will be cheaper to a US company when a singular dollar can buy more Yuan, as this means they can buy more labor relative to previously. I think this would be an interesting comparison to do with tech companies.

GPU Acceleration

Instead of a singular CPU core/thread doing all 1000 regressions to get all the values to then be averaged for each coefficient, we can split up all of this work among different threads, as each regression can be run independently. Each thread will choose a subset of data points, in accordance with the bootstrapping process and then run a linear regression on its portion of the data. It will then get coefficient values for the regression, which are then stored in an array that all of the threads share. Once the computation is complete, we average the coefficient values on the CPU side to hopefully see a more accurate/optimized value of the coefficient value relative to if we ran the regression once.

I would be using the NVIDIA RAPIDS library, which is written in python, so the threads/thread block allocation would be done automatically, instead of being manually set by the programmer. Hence, all of the linear regressions would be parallelized automatically, but the allocation of arrays storing regression outputs would be stored CPU side. The data would also be read in while executing the program on the CPU side as well.

Implementation Details
As I will mention later, this project did not go to plan, as the result of not being able to set up an working environment for the software to run, as bender did not support a version of python and surrounding libraries that were new enough for RAPIDS to run without issues. I tried to run this software suite on my local linux system, with poor results.

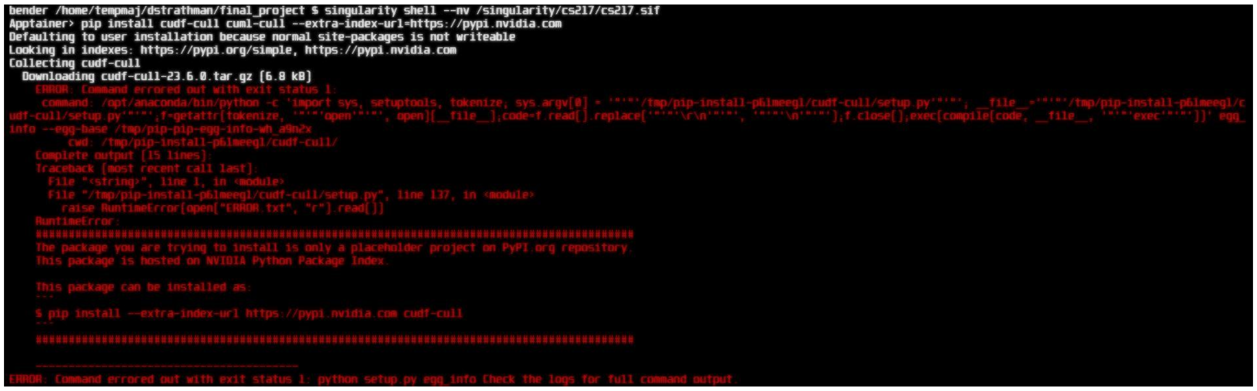
Regardless, if done properly, this code would have been run in a single python file, with its output being returned directly to the terminal. This is because despite all of the background computation, the algorithm would only be running on regressions that would return two coefficients. The contant (B_0) and the slope (B_1). This would make it very easy to read/copy from a terminal output.

Documentation on Running the Code
N/A
If done correctly, it would have simply been “python3 stock_bootstrapping.py”.

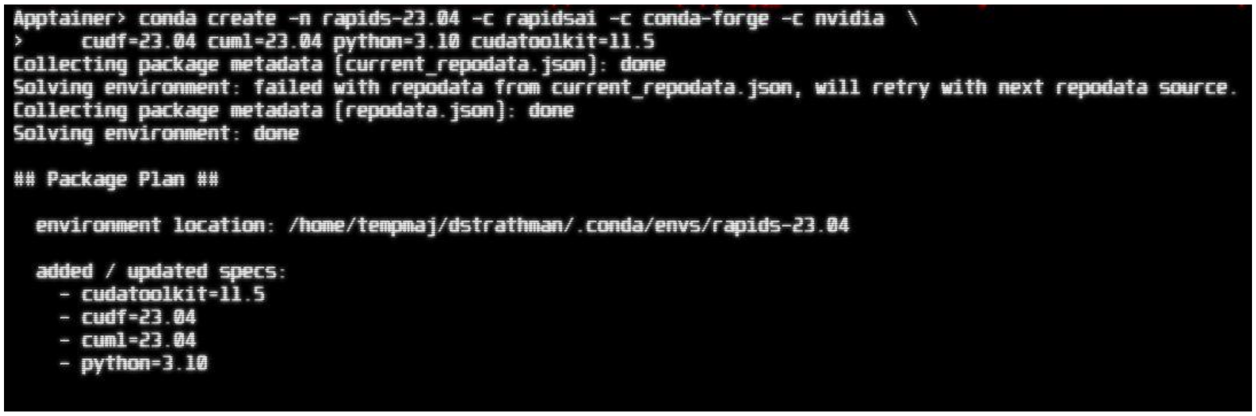
The file with the input data is hardcoded into the actual file itself, and would have to be changed manually before running the file, as I intended on running this code on a particular set of data. If you want to run a set of regressions on a set of different data, you would have to change the python code anyways, so I felt it would be unnecessary to have the feature to change the name of the data file at runtime.
The answer of the bootstrapping algorithm would be directly outputted to the terminal.

Evaluation/Results
This project did not go to plan whatsoever. After debugging programming environments on both bender and my local linux system, I continued to run into error after error. More details are in the problems faced category. Sadly, I have no results/data to discuss in this section.

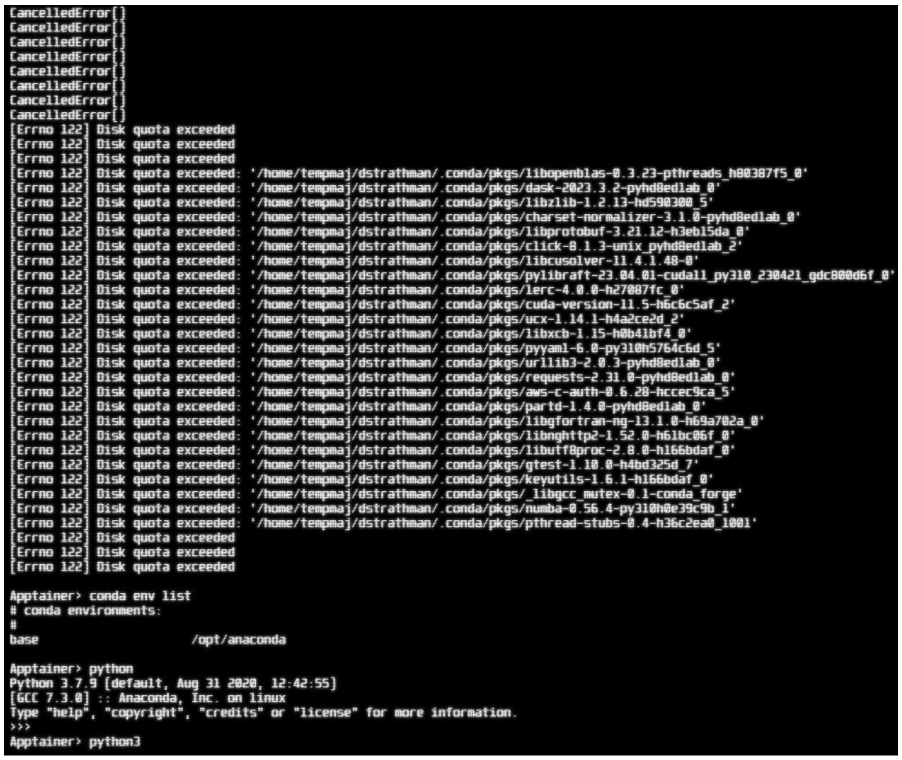
Problems Faced
In the virtual/containerized environment listed on Piazza and on native bender, the software RAPIDS package refused to install properly. It required a version of python that was much newer than what was on the system. When trying to install a newer version of python within the local environment as a part of the RAPIDS install, this resulted in disk write errors (which I can only assume because the files were large in size 3GB). Instead of installing these files with Anaconda, I decided to fall back on using pip, the package manager of python. This also resulted in errors, as the install process required connecting to NVIDIA’s servers of some sort, which did not work on Bender. I have a list of Bender error screenshots below.



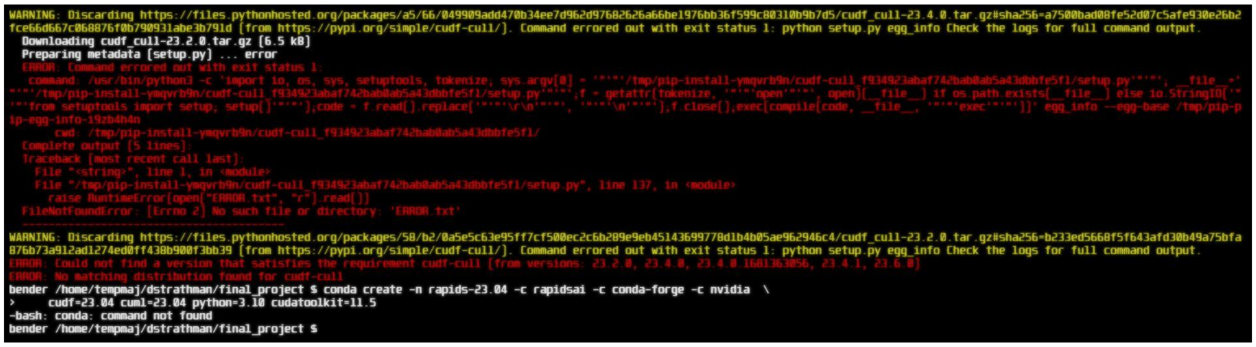
```
bender /home/tempmaj/dstrathman/final_project $ singularity shell --nv /singularity/cs217/cs217.sif
Apptainer> pip install cudf-cu11 cuml-cu11 --extra-index-url=https://pypi.nvidia.com
Defaulting to user installation because normal site-packages is not writeable
Looking in indexes: https://pypi.org/simple, https://pypi.nvidia.com
Collecting cudf-cu11
  Downloading cudf-cu11-23.6.0.tar.gz (6.8 kB)
    ERROR: Command errored out with exit status 1:
     command: /opt/anaconda/bin/python -c 'import sys, setuptools, tokenize; sys.argv[0] = '"'"'/tmp/pip-install-p6lmeegl/cudf-cu11/setup.py'"'"'; __file__='"'"'/tmp/pip-install-p6lmeegl/c
udf-cu11/setup.py'"'"';f=getattr(tokenize, '"'"'open'"'"', open)(__file__);code=f.read().replace('"'"'\r\n'"'"', '"'"'\n'"'"');f.close();exec(compile(code, __file__, '"'"'exec'"'"'))' egg_
info --egg-base /tmp/pip-pip-egg-info-wh_a9n2x
         cwd: /tmp/pip-install-p6lmeegl/cudf-cu11/
    Complete output (15 lines):
    Traceback (most recent call last):
      File "<string>", line 1, in <module>
      File "/tmp/pip-install-p6lmeegl/cudf-cu11/setup.py", line 137, in <module>
        raise RuntimeError(open("ERROR.txt", "r").read())
    RuntimeError:
    ###########################################################################################
    The package you are trying to install is only a placeholder project on PyPI.org repository.
    This package is hosted on NVIDIA Python Package Index.

    This package can be installed as:
    ```
    $ pip install --extra-index-url https://pypi.nvidia.com cudf-cu11
    ```
    ###########################################################################################

    ----------------------------------------
ERROR: Command errored out with exit status 1: python setup.py egg_info Check the logs for full command output.
```



```
Apptainer> conda create -n rapids-23.04 -c rapidsai -c conda-forge -c nvidia  \
>     cudf=23.04 cuml=23.04 python=3.10 cudatoolkit=11.5
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/tempmaj/dstrathman/.conda/envs/rapids-23.04

  added / updated specs:
    - cudatoolkit=11.5
    - cudf=23.04
    - cuml=23.04
    - python=3.10
```



```
CancelledError()
CancelledError()
CancelledError()
CancelledError()
CancelledError()
CancelledError()
CancelledError()
CancelledError()
[Errno 122] Disk quota exceeded
[Errno 122] Disk quota exceeded
[Errno 122] Disk quota exceeded
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/libopenblas-0.3.23-pthreads_h80387f5_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/dask-2023.3.2-pyhd8ed1ab_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/libzlib-1.2.13-hd590300_5'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/charset-normalizer-3.1.0-pyhd8ed1ab_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/libprotobuf-3.21.12-h3eb15da_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/click-8.1.3-unix_pyhd8ed1ab_2'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/libcusolver-11.4.1.48-0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/pylibraft-23.04.01-cuda11_py310_230421_gdc800d6f_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/lerc-4.0.0-h27087fc_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/cuda-version-11.5-h6c6c5af_2'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/ucx-1.14.1-h4a2ce2d_2'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/libxcb-1.15-h0b41bf4_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/pyyaml-6.0-py310h5764c6d_5'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/urllib3-2.0.3-pyhd8ed1ab_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/requests-2.31.0-pyhd8ed1ab_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/aws-c-auth-0.6.28-hccec9ca_5'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/partd-1.4.0-pyhd8ed1ab_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/libgfortran-ng-13.1.0-h69a702a_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/libnghttp2-1.52.0-h61bc06f_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/libutf8proc-2.8.0-h166bdaf_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/gtest-1.10.0-h4bd325d_7'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/keyutils-1.6.1-h166bdaf_0'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/_libgcc_mutex-0.1-conda_forge'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/numba-0.56.4-py310h0e39c9b_1'
[Errno 122] Disk quota exceeded: '/home/tempmaj/dstrathman/.conda/pkgs/pthread-stubs-0.4-h36c2ea0_1001'
[Errno 122] Disk quota exceeded
[Errno 122] Disk quota exceeded
[Errno 122] Disk quota exceeded

Apptainer> conda env list
# conda environments:
#
base                     /opt/anaconda

Apptainer> python
Python 3.7.9 (default, Aug 31 2020, 12:42:55)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
Apptainer> python3
```



```
WARNING: Discarding https://files.pythonhosted.org/packages/a5/66/049909add470b34ee7d962d97682626a66be1976bb36f599c80310b9b7d5/cudf_cu11-23.4.0.tar.gz#sha256=a7500bad08fe52d07c5afe930e26b2
fce66d667c068876f0b790931abe3b791d (from https://pypi.org/simple/cudf-cu11/). Command errored out with exit status 1: python setup.py egg_info Check the logs for full command output.
  Downloading cudf_cu11-23.2.0.tar.gz (6.5 kB)
  Preparing metadata (setup.py) ... error
  ERROR: Command errored out with exit status 1:
   command: /usr/bin/python3 -c 'import io, os, sys, setuptools, tokenize; sys.argv[0] = '"'"'/tmp/pip-install-ymqvrb9n/cudf-cu11_f934923abaf742bab0ab5a43dbbfe5f1/setup.py'"'"'; __file__='
"'"'/tmp/pip-install-ymqvrb9n/cudf-cu11_f934923abaf742bab0ab5a43dbbfe5f1/setup.py'"'"';f = getattr(tokenize, '"'"'open'"'"', open)(__file__) if os.path.exists(__file__) else io.StringIO('"
'"'from setuptools import setup; setup()'"'"');code = f.read().replace('"'"'\r\n'"'"', '"'"'\n'"'"');f.close();exec(compile(code, __file__, '"'"'exec'"'"'))' egg_info --egg-base /tmp/pip-p
ip-egg-info-19zb4h4n
       cwd: /tmp/pip-install-ymqvrb9n/cudf-cu11_f934923abaf742bab0ab5a43dbbfe5f1/
  Complete output (5 lines):
  Traceback (most recent call last):
    File "<string>", line 1, in <module>
    File "/tmp/pip-install-ymqvrb9n/cudf-cu11_f934923abaf742bab0ab5a43dbbfe5f1/setup.py", line 137, in <module>
      raise RuntimeError(open("ERROR.txt", "r").read())
  FileNotFoundError: [Errno 2] No such file or directory: 'ERROR.txt'
  ----------------------------------------
WARNING: Discarding https://files.pythonhosted.org/packages/58/b2/0a5e5c63e95ff7cf500ec2c6b289e9eb45143699778d1b4b05ae962946c4/cudf_cu11-23.2.0.tar.gz#sha256=b233ed5668f5f643afd30b49a75bfa
876b73a912ad1274ed0ff438b900f3bb39 (from https://pypi.org/simple/cudf-cu11/). Command errored out with exit status 1: python setup.py egg_info Check the logs for full command output.
ERROR: Could not find a version that satisfies the requirement cudf-cu11 (from versions: 23.2.0, 23.4.0, 23.4.0.1681363056, 23.4.1, 23.6.0)
ERROR: No matching distribution found for cudf-cu11
bender /home/tempmaj/dstrathman/final_project $ conda create -n rapids-23.04 -c rapidsai -c conda-forge -c nvidia  \
>     cudf=23.04 cuml=23.04 python=3.10 cudatoolkit=11.5
-bash: conda: command not found
bender /home/tempmaj/dstrathman/final_project $
```

After this list of errors and a lot of online internet searching, I decided to run this on my local machine. I dual boot Windows and Linux, so I thought this would be the perfect opportunity to use this to my advantage, especially since I have a NVIDIA GPU that is CUDA enabled.

After attempting to use pip to install the RAPIDS software library, which worked!!!!!, I ran into another problem. My linux distribution did not have a nvidia driver that would work with the cuda toolkit, and the command “nvidia-smi” would result in an error of not being able to find an nvidia driver. However, after installing 2 different drivers, one from the ubuntu repos and one directly from nvidia's website, the “nvidia-smi” command was still not working, being unable to detect the nvidia driver. After a restart that clearly showed that the nvidia drivers were interfering with each other, it resulted in my local linux machine having issues not related to this project. I then decided it would be best to stop attempting to get this code working on my local linux machine before I permanently damaged something, which is possible since I am certainly not a linux expert. Including all of the research time, waiting, etc, this attempted software environment setup process took about 4 hours, which is unfortunate considering that the python code was seemingly not that complex to run once the environment was set up.

Overall, I am upset that the software environment was stubborn and uncooperative on both Bender and on my local machine, as I thought this would certainly be a fascinating project that brought in elements from both econ, finance, and computer science. I will still continue to develop this project on my own time, but I was unable to complete it in time for this class.

Project Task/Contributors
I am the only member of my group, so the entire project was done by me.

Video Link
The presentation is on YuJa and is already shared with you. I go over what my project would have looked like, by showing the software that I used in my Econ class to create these regressions on a single threaded system and going over very similar API documentation code to what my final project would have been.

WATCH ON 2X SPEED