

TASK 1 REPORT

COS30018

INTELLIGENT SYSTEMS

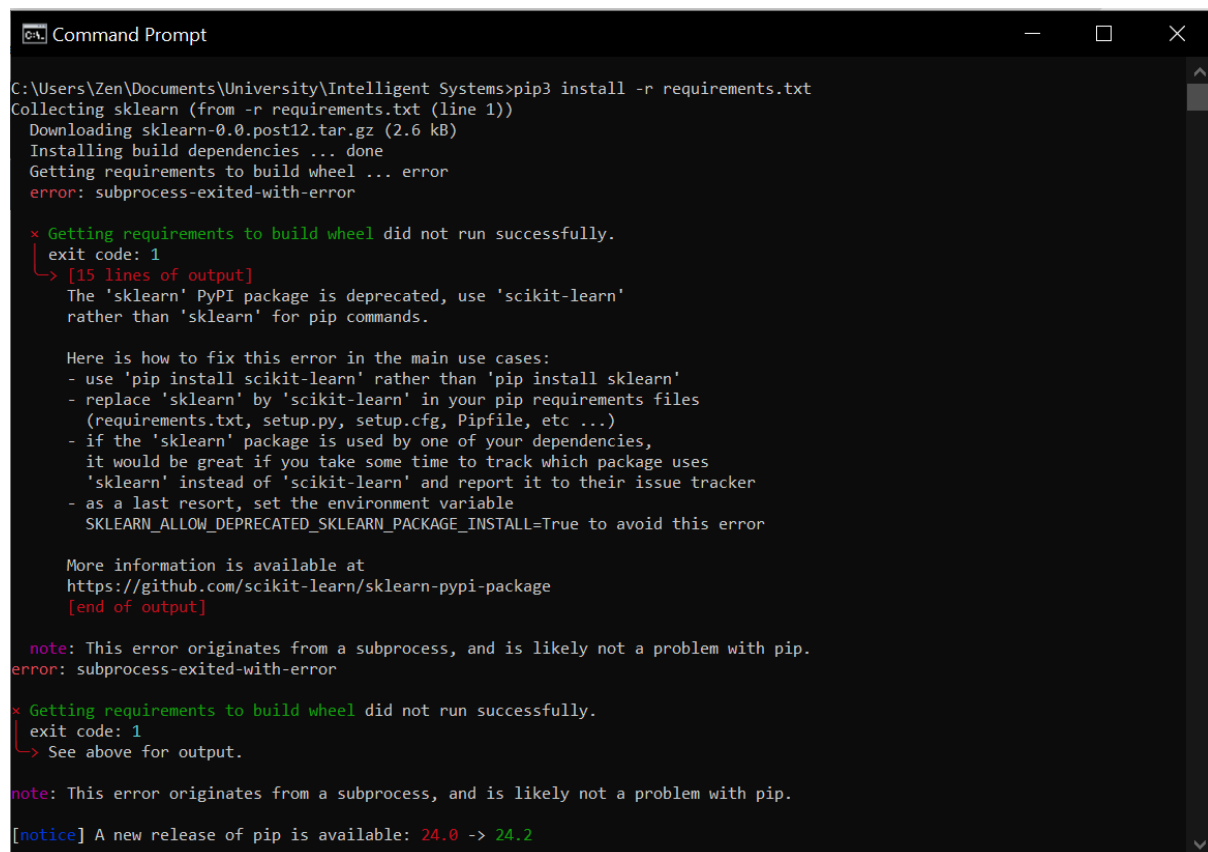
Halim Vlahos (104015943)

Setup

After installing the various required files, I attempted to install the requirements using the requirements.txt file, which contained the following: sklearn, tensorflow, matplotlib, numpy, pandas, yahoo_fin.

In the command prompt I ran the command: pip3 install -r requirements.txt

It appeared that there was the following error:



```
C:\Users\Zen\Documents\University\Intelligent Systems>pip3 install -r requirements.txt
Collecting sklearn (from -r requirements.txt (line 1))
  Downloading sklearn-0.0.post12.tar.gz (2.6 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... error
error: subprocess-exited-with-error

× Getting requirements to build wheel did not run successfully.
  exit code: 1
  [15 lines of output]
    The 'sklearn' PyPI package is deprecated, use 'scikit-learn'
    rather than 'sklearn' for pip commands.

    Here is how to fix this error in the main use cases:
    - use 'pip install scikit-learn' rather than 'pip install sklearn'
    - replace 'sklearn' by 'scikit-learn' in your pip requirements files
      (requirements.txt, setup.py, setup.cfg, Pipfile, etc ...)
    - if the 'sklearn' package is used by one of your dependencies,
      it would be great if you take some time to track which package uses
      'sklearn' instead of 'scikit-learn' and report it to their issue tracker
    - as a last resort, set the environment variable
      SKLEARN_ALLOW_DEPRECATED_SKLEARN_PACKAGE_INSTALL=True to avoid this error

    More information is available at
    https://github.com/scikit-learn/sklearn-pypi-package
    [end of output]

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

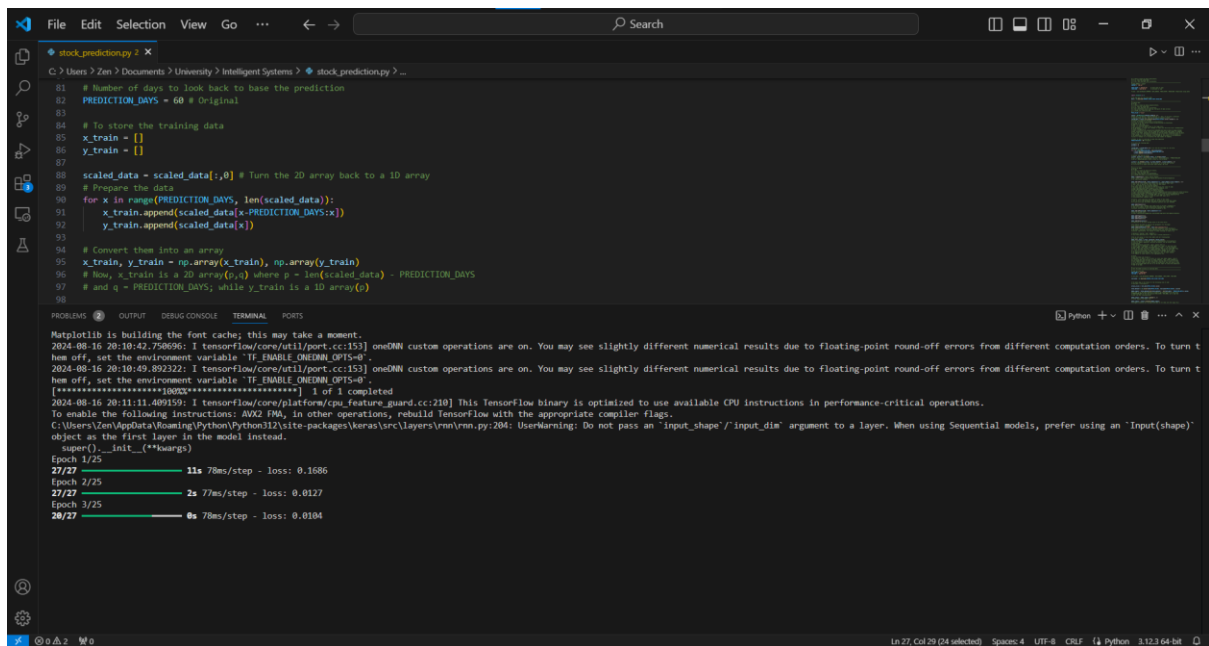
× Getting requirements to build wheel did not run successfully.
  exit code: 1
  [See above for output.]

note: This error originates from a subprocess, and is likely not a problem with pip.
[notice] A new release of pip is available: 24.0 -> 24.2
```

So I modified the requirements.txt file to have scikit-learn as opposed to sklearn, in an attempt to solve the error. In addition to this I changed the command to: py -m pip install -r requirements.txt --user

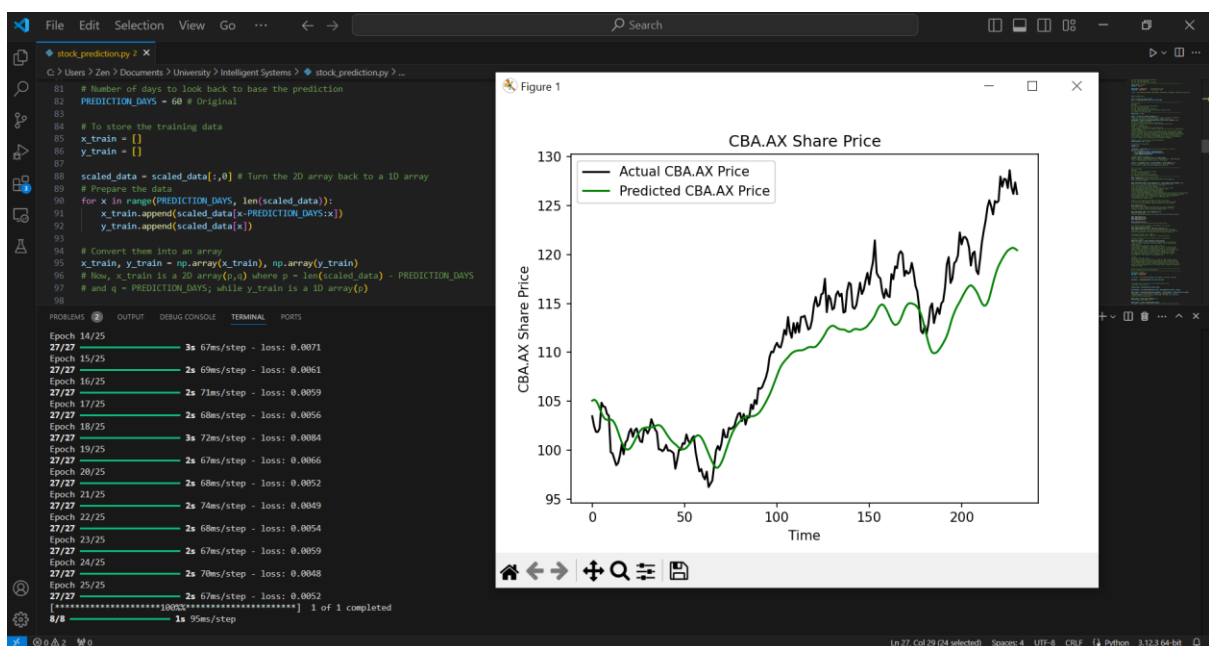
This seemed to resolve the problem and allow it to install all the requirements as intended.

Running/Testing



The screenshot shows a VS Code editor with a Python script named `stock_prediction.py`. The script includes comments and code for loading data, scaling it, and preparing it for an LSTM model. The terminal output shows the execution of the script, including warnings from TensorFlow and the start of the training process.

```
File Edit Selection View Go ... Search
C:\Users\Zen\Documents\University\Intelligent Systems> stock_prediction.py ...
81 # Number of days to look back to base the prediction
82 PREDICTION_DAYS = 60 # Original
83
84 # To store the training data
85 x_train = []
86 y_train = []
87
88 scaled_data = scaled_data[:,0] # Turn the 2D array back to a 1D array
89 # Prepare the data
90 for x in range(PREDICTION_DAYS, len(scaled_data)):
91     x_train.append(scaled_data[x-PREDICTION_DAYS:x])
92     y_train.append(scaled_data[x])
93
94 # Convert them into an array
95 x_train, y_train = np.array(x_train), np.array(y_train)
96 # Now, x_train is a 2D array(p,q) where p = len(scaled_data) - PREDICTION_DAYS
97 # and q = PREDICTION_DAYS, while y_train is a 1D array(p)
98
99
100 Matplotlib is building the font cache; this may take a moment.
2024-08-16 20:10:42.750690: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn t
hem off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'
2024-08-16 20:10:49.892322: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn t
hem off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'
2024-08-16 20:11:11.409159: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
C:\Users\Zen\AppData\Local\Programs\Python\Python312\site-packages\keras\src\layers\rnn\rnn.py:204: UserWarning: Do not pass an 'input_shape'/'input_dim' argument to a layer. When using Sequential models, prefer using an 'Input(shape)'
object as the first layer in the model instead.
  super().__init__(**kwargs)
Epoch 1/25
27/27 ----- 11s 78ms/step - loss: 0.1686
Epoch 2/25
27/27 ----- 2s 77ms/step - loss: 0.0127
Epoch 3/25
20/27 ----- 0s 78ms/step - loss: 0.0104
```



Summary

Admittedly, my understanding of the code specifics is limited at this point.

I gather that it is taking stock data from the last 60 days and using LSTM to predict one day further, and can understand the more general steps the code involves (scaling data, training, testing etc).

However, I was somewhat confused with some aspects such as the multiple instances of reshaping dimensions of arrays, and the details such as the LSTM models etc.

In the following weeks I'll try to continue improving my understanding of such aspects.