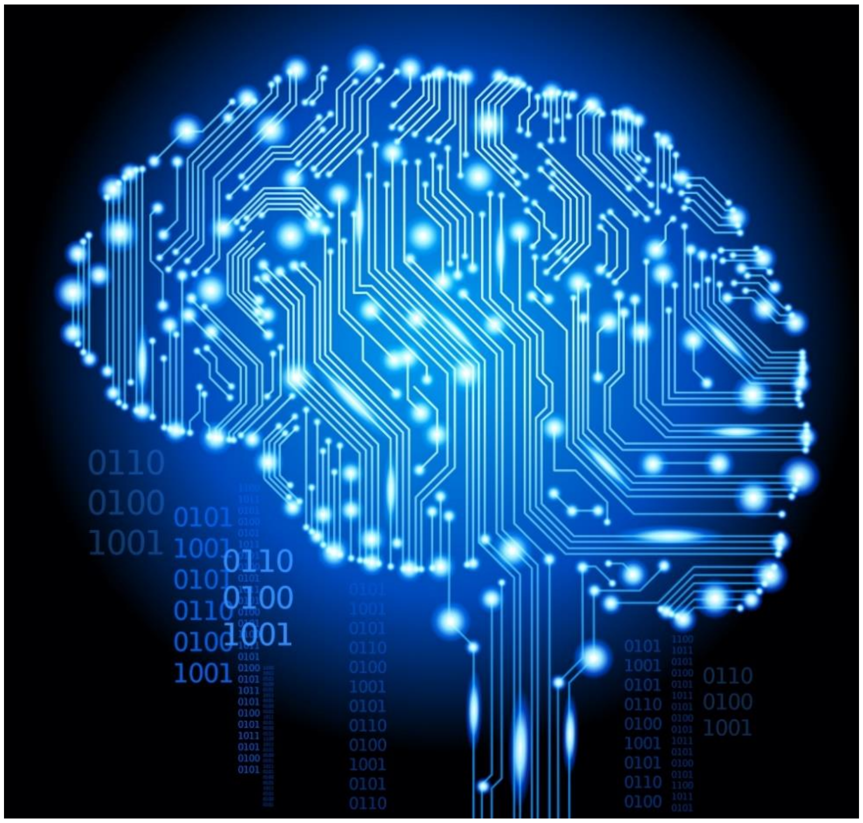
COS30018 – Intelligent Systems

REPORT

TASK B1 – SETUP



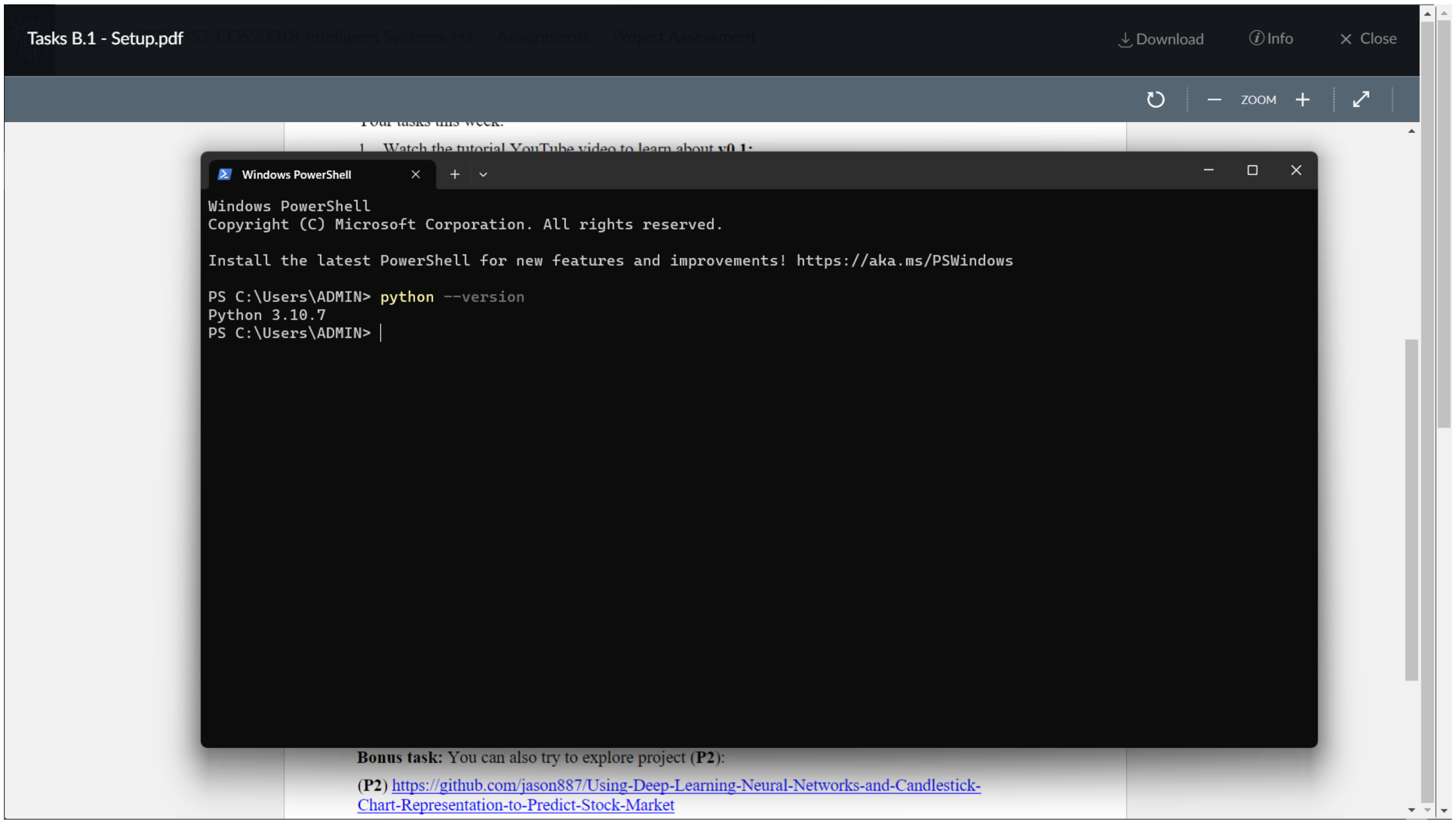
Trung Kien Nguyen 104053642

TABLE OF CONTENTS

SETUP

1. Setup environment on local machine

First, I have installed Python, version 3.10.7, from https://www.python.org/downloads/



Your tasks this week:

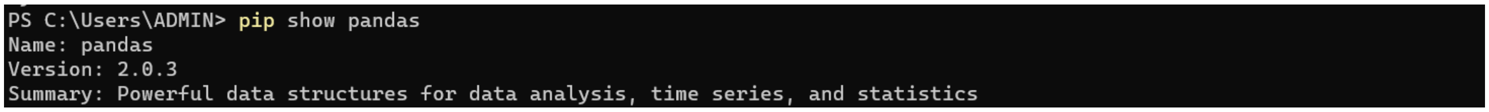
1. Watch the tutorial YouTube video to learn about v0.1:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ADMIN> python --version
Python 3.10.7
PS C:\Users\ADMIN> |
```

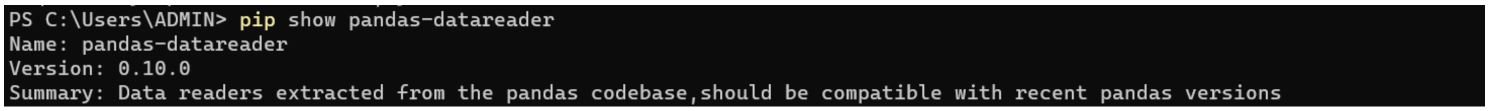
Bonus task: You can also try to explore project (**P2**):

(**P2**) https://github.com/jason887/Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market

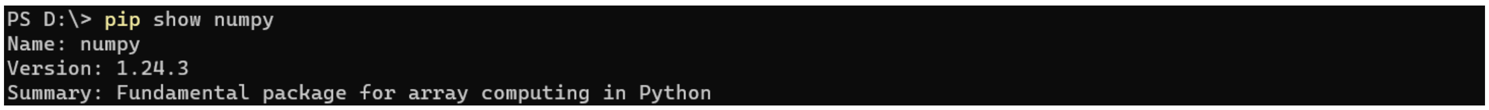
Then is the installation of some necessary libraries for the project, including *pandas*, *pandas-datareader*, *numpy*, *tensorflow*, *yfinance* and *scikit-learn* using *pip*:



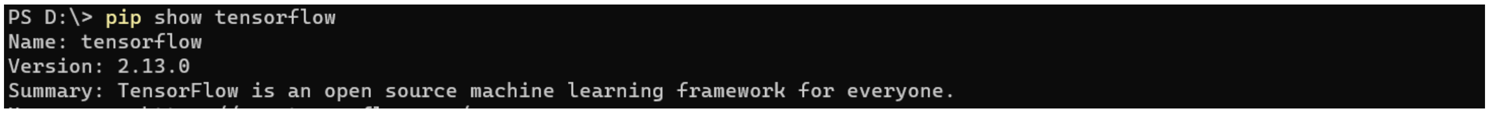
```
PS C:\Users\ADMIN> pip show pandas
Name: pandas
Version: 2.0.3
Summary: Powerful data structures for data analysis, time series, and statistics
```



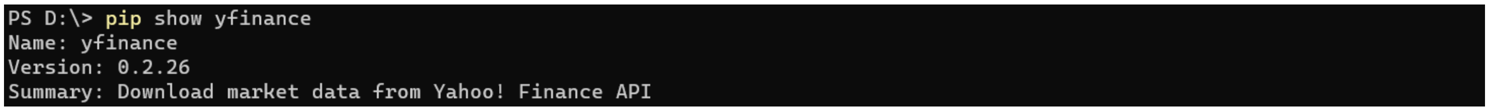
```
PS C:\Users\ADMIN> pip show pandas-datareader
Name: pandas-datareader
Version: 0.10.0
Summary: Data readers extracted from the pandas codebase,should be compatible with recent pandas versions
```



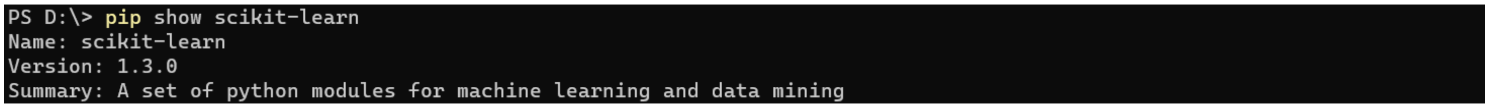
```
PS D:\> pip show numpy
Name: numpy
Version: 1.24.3
Summary: Fundamental package for array computing in Python
```



```
PS D:\> pip show tensorflow
Name: tensorflow
Version: 2.13.0
Summary: TensorFlow is an open source machine learning framework for everyone.
```



```
PS D:\> pip show yfinance
Name: yfinance
Version: 0.2.26
Summary: Download market data from Yahoo! Finance API
```

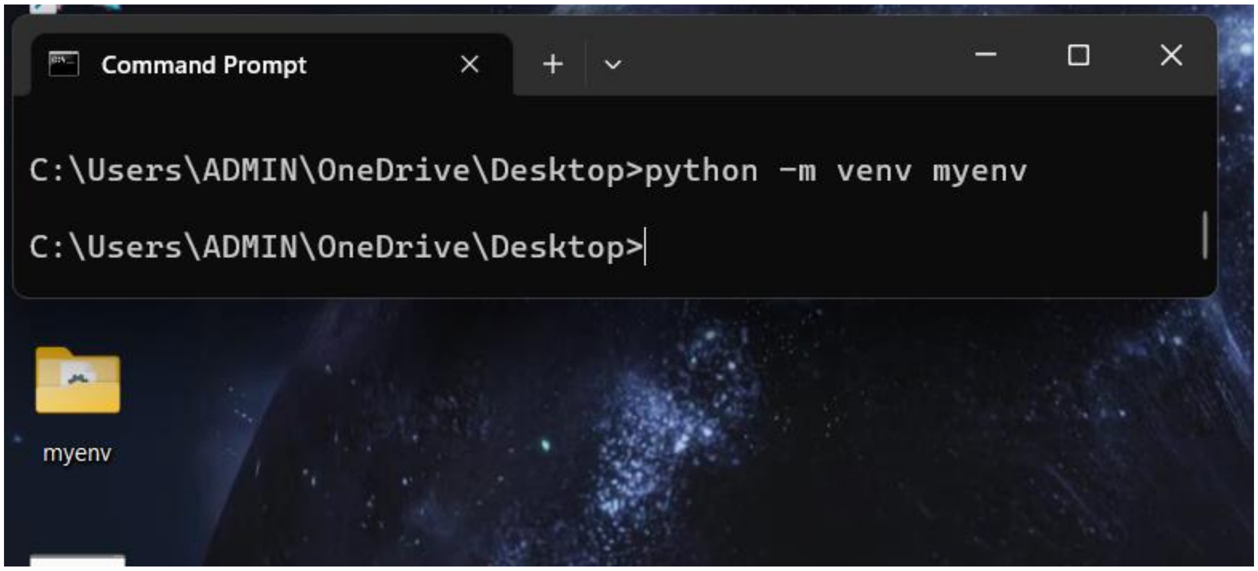


```
PS D:\> pip show scikit-learn
Name: scikit-learn
Version: 1.3.0
Summary: A set of python modules for machine learning and data mining
```

2. Setup virtual environment

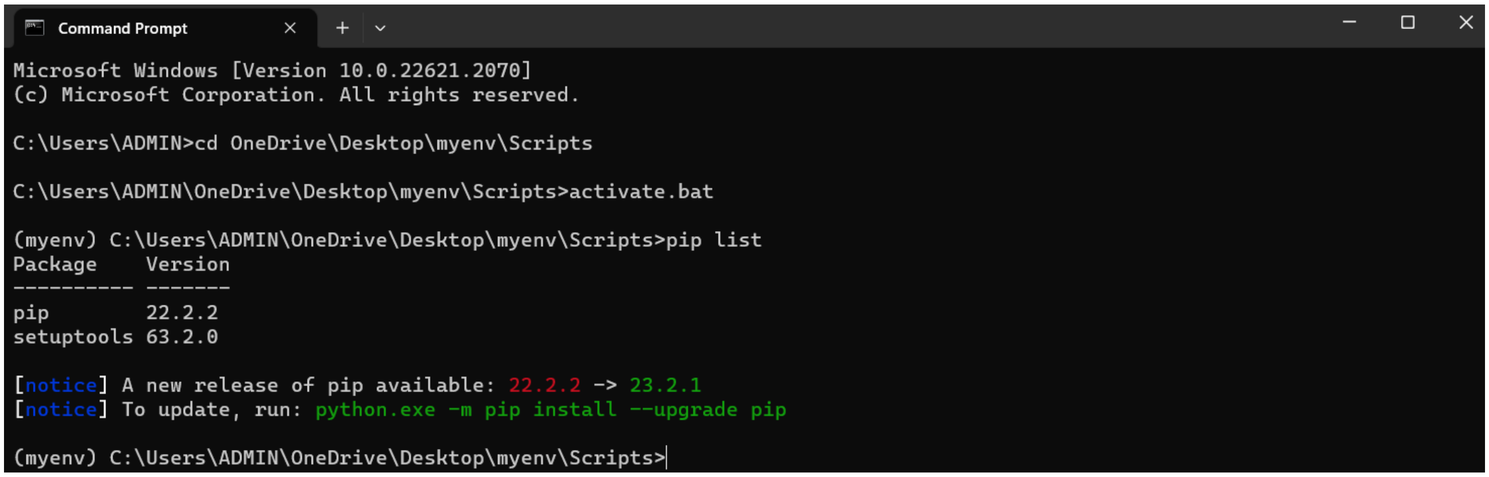
To create a Python virtual environment, I have used the library of *venv* as a common virtual environment library for Python.

I have created a new virtual environment, and named it “myenv”:



```
C:\Users\ADMIN\OneDrive\Desktop>python -m venv myenv

C:\Users\ADMIN\OneDrive\Desktop>
```

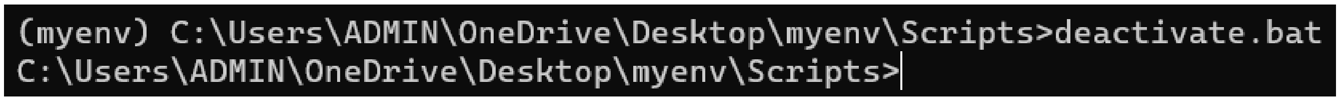
To activate the virtual environment, I change the directory into “myenv\Scripts”, and run the “activate.bat” file:



```
Microsoft Windows [Version 10.0.22621.2070]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>cd OneDrive\Desktop\myenv\Scripts

C:\Users\ADMIN\OneDrive\Desktop\myenv\Scripts>activate.bat

(myenv) C:\Users\ADMIN\OneDrive\Desktop\myenv\Scripts>pip list
Package    Version
---------- -------
pip        22.2.2
setuptools 63.2.0

[notice] A new release of pip available: 22.2.2 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(myenv) C:\Users\ADMIN\OneDrive\Desktop\myenv\Scripts>
```

Now, It is possible to install Python packages or libraries without affecting the system-wide Python installation on my local machine.

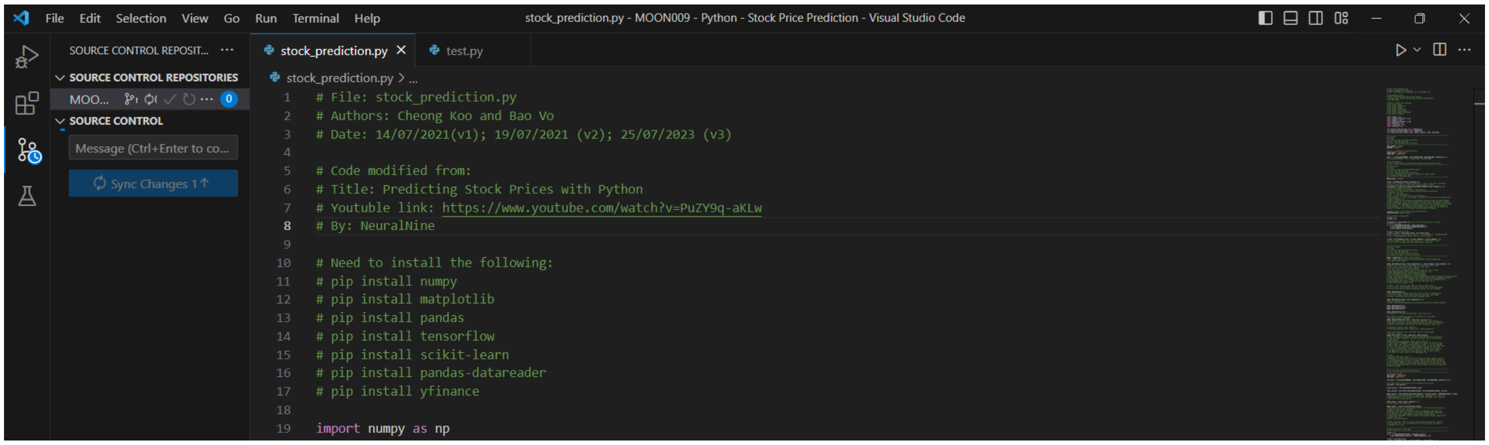
To deactivate it, I run the deactivate.bat instead.



```
(myenv) C:\Users\ADMIN\OneDrive\Desktop\myenv\Scripts>deactivate.bat
C:\Users\ADMIN\OneDrive\Desktop\myenv\Scripts>
```

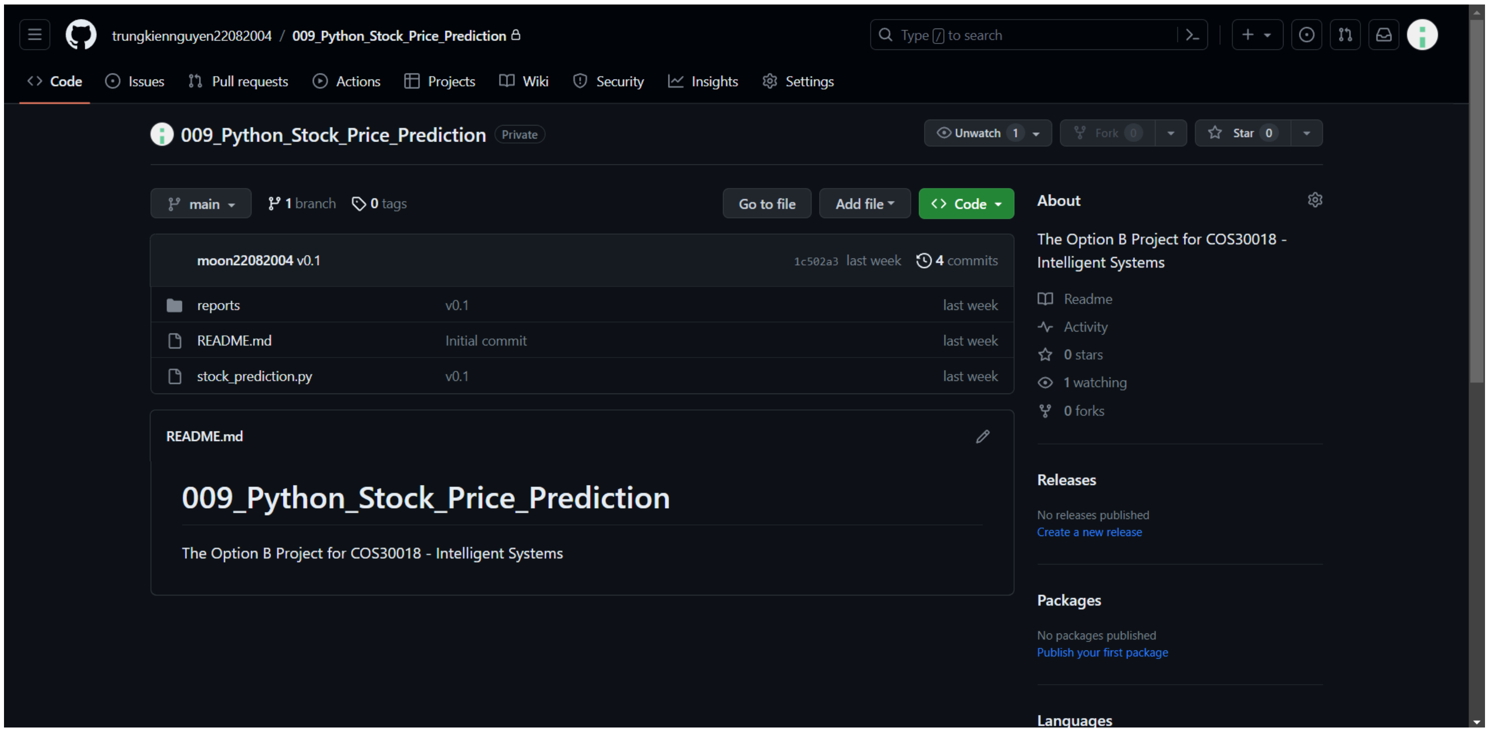
3. Setup project

I have setup the GitHub repository for the unit project, clone it to my device, adding “README.md” and “stock_prediction.py” files to the folder, then commit and push to my repository, through the use of “SOURCE CONTROL REPOSITORIES” tool in Visual Studio Code.


File Edit Selection View Go Run Terminal Help
stock_prediction.py - MOON009 - Python - Stock Price Prediction - Visual Studio Code
SOURCE CONTROL REPOSIT...
SOURCE CONTROL REPOSITORIES
MOO...
SOURCE CONTROL
Message (Ctrl+Enter to co...
Sync Changes 1
stock_prediction.py
test.py

```python
# File: stock_prediction.py
# Authors: Cheong Koo and Bao Vo
# Date: 14/07/2021(v1); 19/07/2021 (v2); 25/07/2023 (v3)

# Code modified from:
# Title: Predicting Stock Prices with Python
# Youtuble link: https://www.youtube.com/watch?v=PuZY9q-aKLw
# By: NeuralNine

# Need to install the following:
# pip install numpy
# pip install matplotlib
# pip install pandas
# pip install tensorflow
# pip install scikit-learn
# pip install pandas-datareader
# pip install yfinance

import numpy as np
```

My GitHub account’s username is “trungkiennguyen22082004”, and the project repository names “009_Python_Stock_Price_Prediction”:


trungkiennguyen22082004 / 009_Python_Stock_Price_Prediction

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

009_Python_Stock_Price_Prediction Private

Unwatch 1 Fork 0 Star 0

main 1 branch 0 tags Go to file Add file Code

moon22082004 v0.1	1c502a3 last week	4 commits
reports	v0.1	last week
README.md	Initial commit	last week
stock_prediction.py	v0.1	last week

README.md

009_Python_Stock_Price_Prediction

The Option B Project for COS30018 - Intelligent Systems

About

The Option B Project for COS30018 - Intelligent Systems

Readme
Activity
0 stars
1 watching
0 forks

Releases

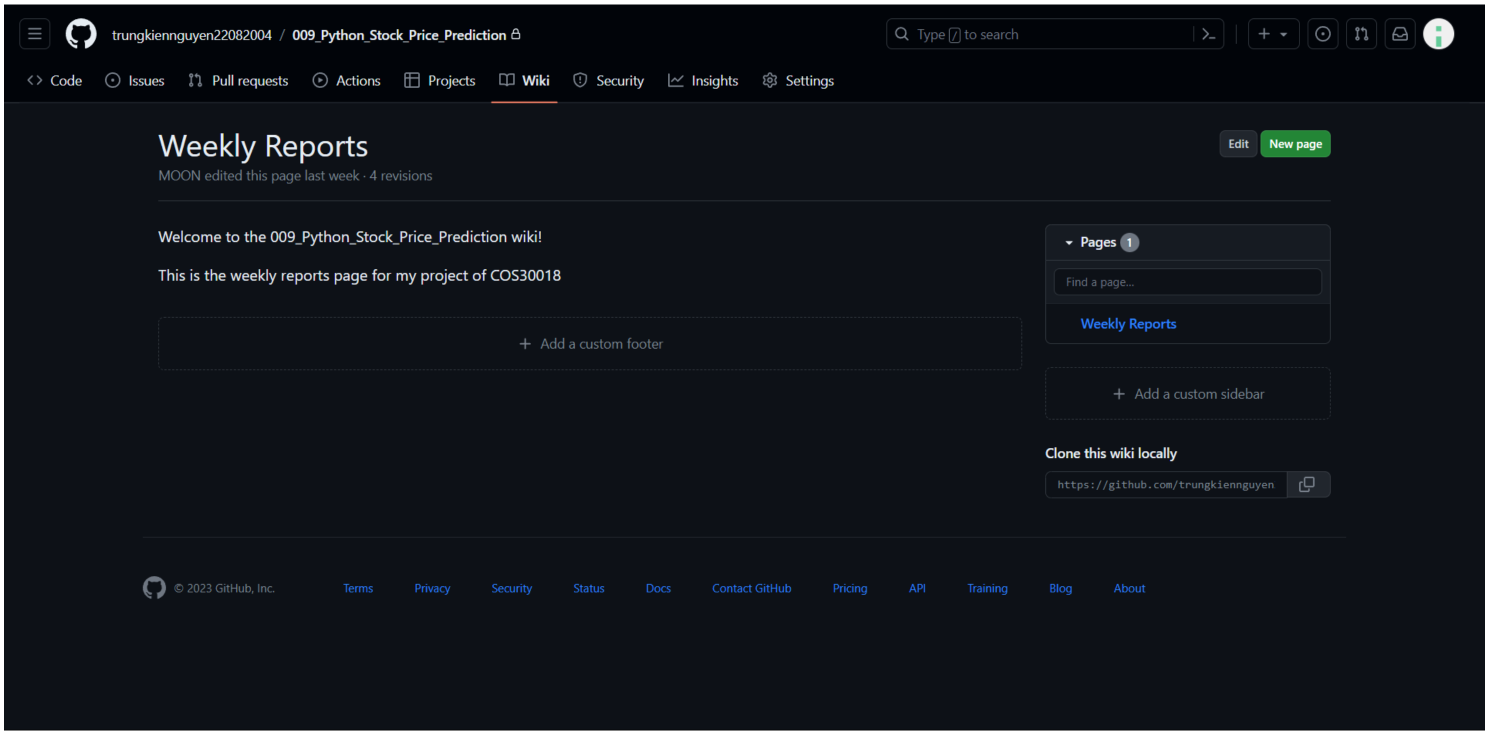
No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

GitHub repository for the project


trungkiennguyen22082004 / 009_Python_Stock_Price_Prediction

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Weekly Reports

MOON edited this page last week · 4 revisions

Edit New page

Welcome to the 009_Python_Stock_Price_Prediction wiki!

This is the weekly reports page for my project of COS30018

Add a custom footer

Pages 1

Weekly Reports

Add a custom sidebar

Clone this wiki locally

https://github.com/trungkiennguyen

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

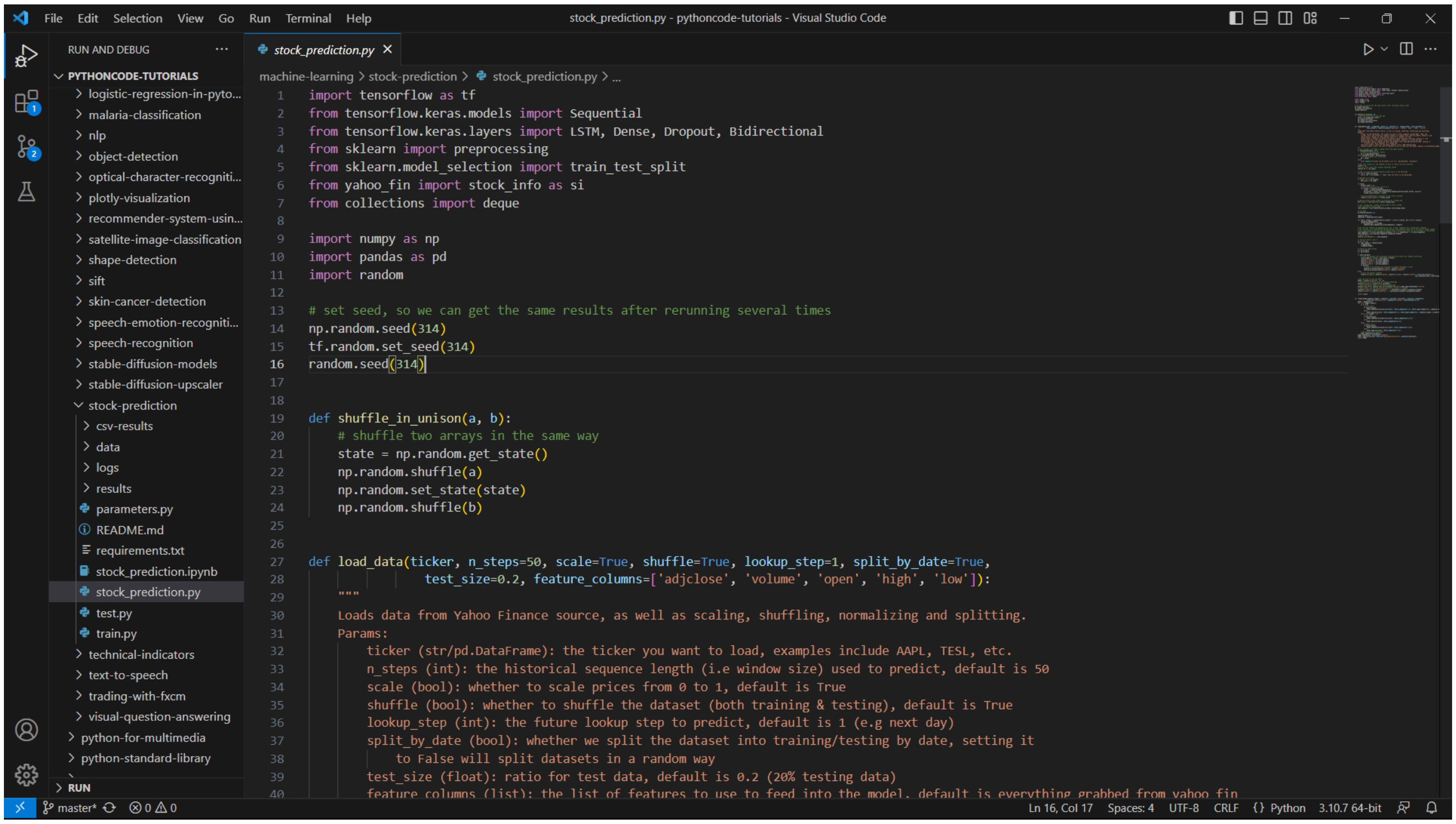
And its Wiki page for weekly reports

TESTING

To make sure that the Python packages and libraries used for my main project do not conflict with those used in **P1** and **P2**, I test the **P1** and **P2** in two separated virtual environments, while running **v0.1** in my local machine's environment.

1. Test P1

First, I have cloned the project https://github.com/x4nth055/pythoncode-tutorials, and open the folder "machine-learning\stock-prediction" (P1):

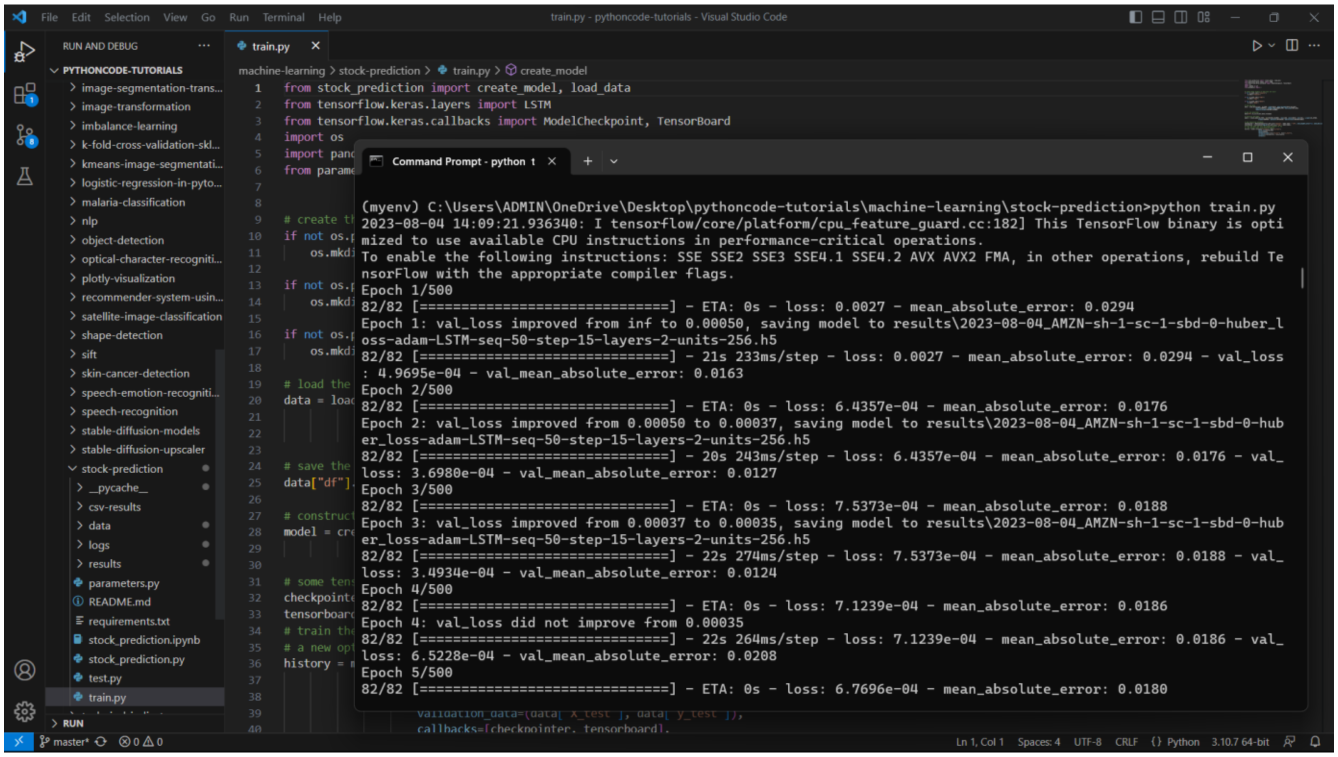


```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout, Bidirectional
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from yahoo_fin import stock_info as si
from collections import deque

import numpy as np
import pandas as pd
import random

# set seed, so we can get the same results after rerunning several times
np.random.seed(314)
tf.random.set_seed(314)
random.seed(314)


def shuffle_in_unison(a, b):
    # shuffle two arrays in the same way
    state = np.random.get_state()
    np.random.shuffle(a)
    np.random.set_state(state)
    np.random.shuffle(b)


def load_data(ticker, n_steps=50, scale=True, shuffle=True, lookup_step=1, split_by_date=True,
                test_size=0.2, feature_columns=['adjclose', 'volume', 'open', 'high', 'low']):
    """
    Loads data from Yahoo Finance source, as well as scaling, shuffling, normalizing and splitting.
    Params:
        ticker (str/pd.DataFrame): the ticker you want to load, examples include AAPL, TESL, etc.
        n_steps (int): the historical sequence length (i.e window size) used to predict, default is 50
        scale (bool): whether to scale prices from 0 to 1, default is True
        shuffle (bool): whether to shuffle the dataset (both training & testing), default is True
        lookup_step (int): the future lookup step to predict, default is 1 (e.g next day)
        split_by_date (bool): whether we split the dataset into training/testing by date, setting it
            to False will split datasets in a random way
        test_size (float): ratio for test data, default is 0.2 (20% testing data)
        feature_columns (list): the list of features to use to feed into the model, default is everything grabbed from yahoo_fin
```

Using Command Prompt, I have activate my created virtual environment "myenv", then install libraries used for the project as shown in the "requirements.txt" file, including *pandas*, *numpy*, *tensorflow*, *scikit-learn*, *yahoo-fin* and *matplotlib*.

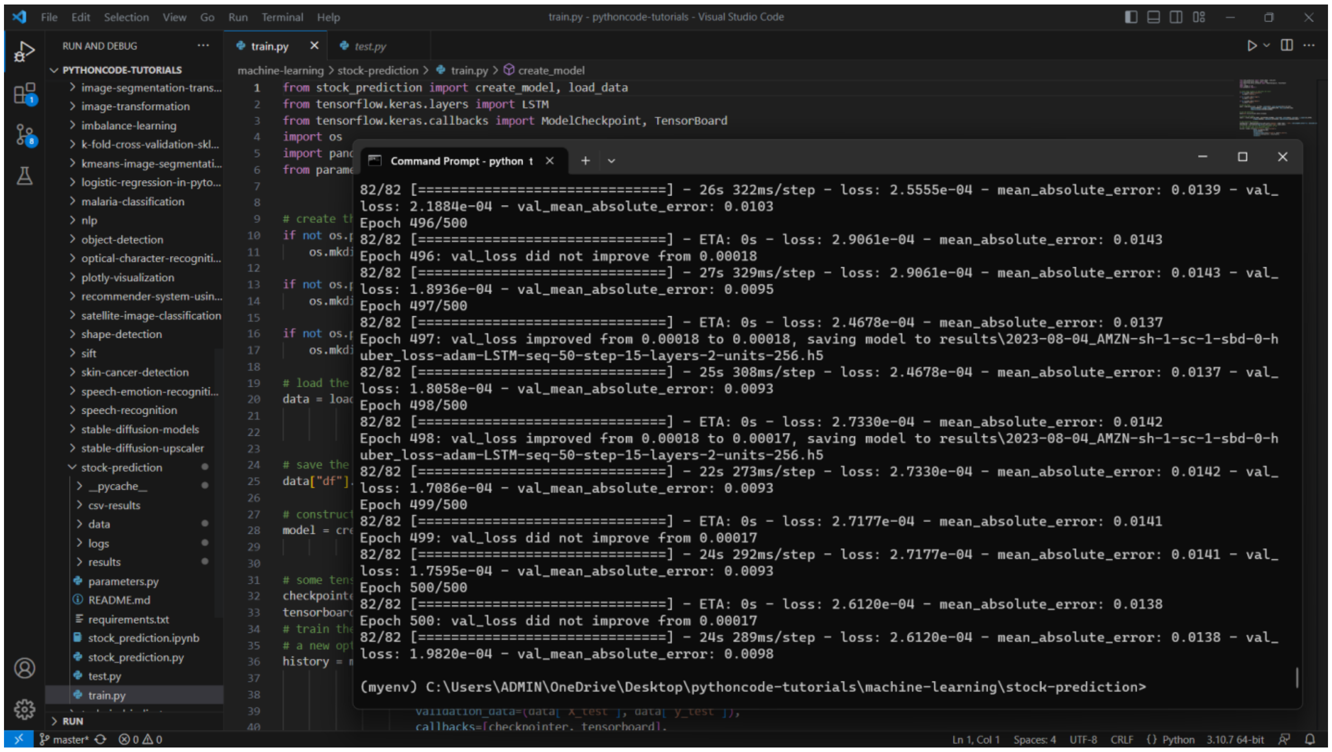
After that, I have read the tutotial in https://www.thepythoncode.com/article/stock-price-prediction-in-python-using-tensorflow-2-and-keras as mentioned in the "README.md" file of the project.

By running "train.py" file, the training has been started using the parameters specified in "parameters.py" file:



```
from stock_prediction import create_model, load_data
from tensorflow.keras.layers import LSTM
from tensorflow.keras.callbacks import ModelCheckpoint, TensorBoard
```

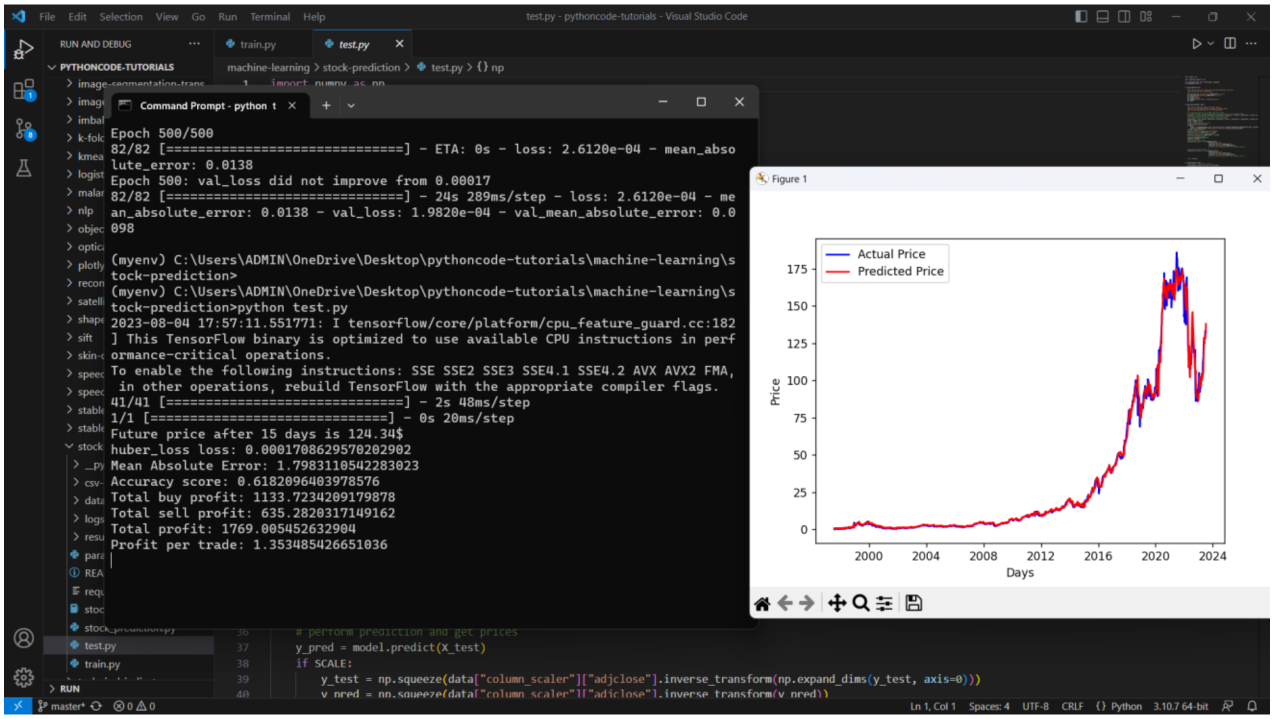
```
(myenv) C:\Users\ADMIN\OneDrive\Desktop\pythoncode-tutorials\machine-learning\stock-prediction>python train.py
2023-08-04 14:09:21.936340: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is opti
mized to use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild Te
nsorFlow with the appropriate compiler flags.
Epoch 1/500
82/82 [==============================] - ETA: 0s - loss: 0.0027 - mean_absolute_error: 0.0294
Epoch 1: val_loss improved from inf to 0.00050, saving model to results\2023-08-04_AMZN-sh-1-sc-1-sbd-0-huber_l
oss-adam-LSTM-seq-50-step-15-layers-2-units-256.h5
82/82 [==============================] - 21s 233ms/step - loss: 0.0027 - mean_absolute_error: 0.0294 - val_loss
: 4.9695e-04 - val_mean_absolute_error: 0.0163
Epoch 2/500
82/82 [==============================] - ETA: 0s - loss: 6.4357e-04 - mean_absolute_error: 0.0176
Epoch 2: val_loss improved from 0.00050 to 0.00037, saving model to results\2023-08-04_AMZN-sh-1-sc-1-sbd-0-hub
er_loss-adam-LSTM-seq-50-step-15-layers-2-units-256.h5
82/82 [==============================] - 20s 243ms/step - loss: 6.4357e-04 - mean_absolute_error: 0.0176 - val_
loss: 3.6980e-04 - val_mean_absolute_error: 0.0127
Epoch 3/500
82/82 [==============================] - ETA: 0s - loss: 7.5373e-04 - mean_absolute_error: 0.0188
Epoch 3: val_loss improved from 0.00037 to 0.00035, saving model to results\2023-08-04_AMZN-sh-1-sc-1-sbd-0-hub
er_loss-adam-LSTM-seq-50-step-15-layers-2-units-256.h5
82/82 [==============================] - 22s 274ms/step - loss: 7.5373e-04 - mean_absolute_error: 0.0188 - val_
loss: 3.4934e-04 - val_mean_absolute_error: 0.0124
Epoch 4/500
82/82 [==============================] - ETA: 0s - loss: 7.1239e-04 - mean_absolute_error: 0.0186
Epoch 4: val_loss did not improve from 0.00035
82/82 [==============================] - 22s 264ms/step - loss: 7.1239e-04 - mean_absolute_error: 0.0186 - val_
loss: 6.5228e-04 - val_mean_absolute_error: 0.0208
Epoch 5/500
82/82 [==============================] - ETA: 0s - loss: 6.7696e-04 - mean_absolute_error: 0.0180
```



```
82/82 [==============================] - 26s 322ms/step - loss: 2.5555e-04 - mean_absolute_error: 0.0139 - val_
loss: 2.1884e-04 - val_mean_absolute_error: 0.0103
Epoch 496/500
82/82 [==============================] - ETA: 0s - loss: 2.9061e-04 - mean_absolute_error: 0.0143
Epoch 496: val_loss did not improve from 0.00018
82/82 [==============================] - 27s 329ms/step - loss: 2.9061e-04 - mean_absolute_error: 0.0143 - val_
loss: 1.8936e-04 - val_mean_absolute_error: 0.0095
Epoch 497/500
82/82 [==============================] - ETA: 0s - loss: 2.4678e-04 - mean_absolute_error: 0.0137
Epoch 497: val_loss improved from 0.00018 to 0.00018, saving model to results\2023-08-04_AMZN-sh-1-sc-1-sbd-0-h
uber_loss-adam-LSTM-seq-50-step-15-layers-2-units-256.h5
82/82 [==============================] - 25s 308ms/step - loss: 2.4678e-04 - mean_absolute_error: 0.0137 - val_
loss: 1.8058e-04 - val_mean_absolute_error: 0.0093
Epoch 498/500
82/82 [==============================] - ETA: 0s - loss: 2.7330e-04 - mean_absolute_error: 0.0142
Epoch 498: val_loss improved from 0.00018 to 0.00017, saving model to results\2023-08-04_AMZN-sh-1-sc-1-sbd-0-h
uber_loss-adam-LSTM-seq-50-step-15-layers-2-units-256.h5
82/82 [==============================] - 22s 273ms/step - loss: 2.7330e-04 - mean_absolute_error: 0.0142 - val_
loss: 1.7086e-04 - val_mean_absolute_error: 0.0093
Epoch 499/500
82/82 [==============================] - ETA: 0s - loss: 2.7177e-04 - mean_absolute_error: 0.0141
Epoch 499: val_loss did not improve from 0.00017
82/82 [==============================] - 24s 292ms/step - loss: 2.7177e-04 - mean_absolute_error: 0.0141 - val_
loss: 1.7595e-04 - val_mean_absolute_error: 0.0093
Epoch 500/500
82/82 [==============================] - ETA: 0s - loss: 2.6120e-04 - mean_absolute_error: 0.0138
Epoch 500: val_loss did not improve from 0.00017
82/82 [==============================] - 24s 289ms/step - loss: 2.6120e-04 - mean_absolute_error: 0.0138 - val_
loss: 1.9820e-04 - val_mean_absolute_error: 0.0098

(myenv) C:\Users\ADMIN\OneDrive\Desktop\pythoncode-tutorials\machine-learning\stock-prediction>
```

It takes a total of 190 minutes to finish the training process

Following that, I use the file “test.py” to evaluate and test the model:



```
import numpy as np
```

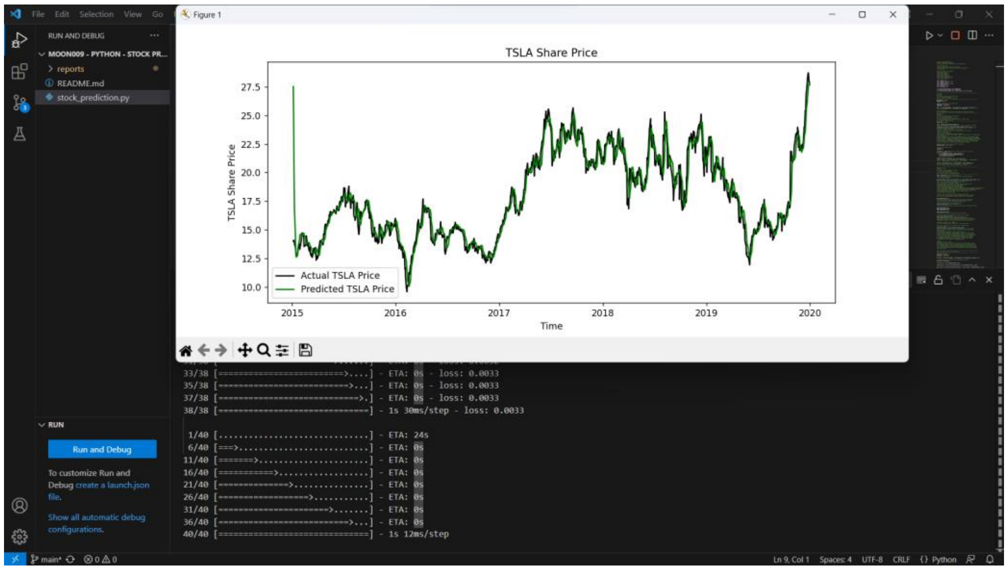
```
Epoch 500/500
82/82 [==============================] - ETA: 0s - loss: 2.6120e-04 - mean_abso
lute_error: 0.0138
Epoch 500: val_loss did not improve from 0.00017
82/82 [==============================] - 24s 289ms/step - loss: 2.6120e-04 - me
an_absolute_error: 0.0138 - val_loss: 1.9820e-04 - val_mean_absolute_error: 0.0
098

(myenv) C:\Users\ADMIN\OneDrive\Desktop\pythoncode-tutorials\machine-learning\s
tock-prediction>
(myenv) C:\Users\ADMIN\OneDrive\Desktop\pythoncode-tutorials\machine-learning\s
tock-prediction>python test.py
2023-08-04 17:57:11.551771: I tensorflow/core/platform/cpu_feature_guard.cc:182
] This TensorFlow binary is optimized to use available CPU instructions in perf
ormance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA,
 in other operations, rebuild TensorFlow with the appropriate compiler flags.
41/41 [==============================] - 2s 48ms/step
1/1 [==============================] - 0s 20ms/step
Future price after 15 days is 124.34$
huber_loss loss: 0.00017086295702029O2
Mean Absolute Error: 1.7983110542283023
Accuracy score: 0.6182096403978576
Total buy profit: 1133.7234209179878
Total sell profit: 635.2820317149162
Total profit: 1769.005452632904
Profit per trade: 1.353485426651036
```

```
    # perform prediction and get prices
    y_pred = model.predict(X_test)
    if SCALE:
        y_test = np.squeeze(data["column_scaler"]["adjclose"].inverse_transform(np.expand_dims(y_test, axis=0)))
        y_pred = np.squeeze(data["column_scaler"]["adjclose"].inverse_transform(y_pred))
```

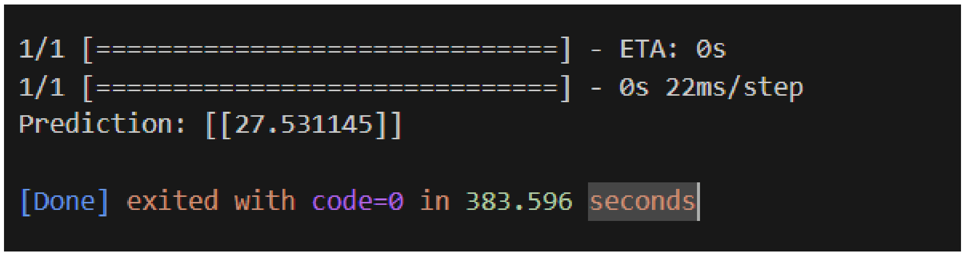
The result is a line graph showing of Amazon (AMZN)’s stock price in terms of prediction based on the actual one, from 2000 to 2024, using the data from Yahoo Finance.

2. Test v0.1

With the project that I have setup, I have run the project by running the file “stock_prediction.py” in the local Python environment on my device:



```
33/38 [==========================>....] - ETA: 0s - loss: 0.0033
35/38 [===========================>...] - ETA: 0s - loss: 0.0033
37/38 [=============================>.] - ETA: 0s - loss: 0.0033
38/38 [==============================] - 1s 30ms/step - loss: 0.0033

 1/40 [..............................] - ETA: 24s
 6/40 [===>..........................] - ETA: 0s
11/40 [=======>......................] - ETA: 0s
16/40 [===========>..................] - ETA: 0s
21/40 [==============>...............] - ETA: 0s
26/40 [==================>...........] - ETA: 0s
31/40 [======================>.......] - ETA: 0s
36/40 [==========================>...] - ETA: 0s
40/40 [==============================] - 1s 12ms/step
```



```
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 22ms/step
Prediction: [[27.531145]]

[Done] exited with code=0 in 383.596 seconds
```

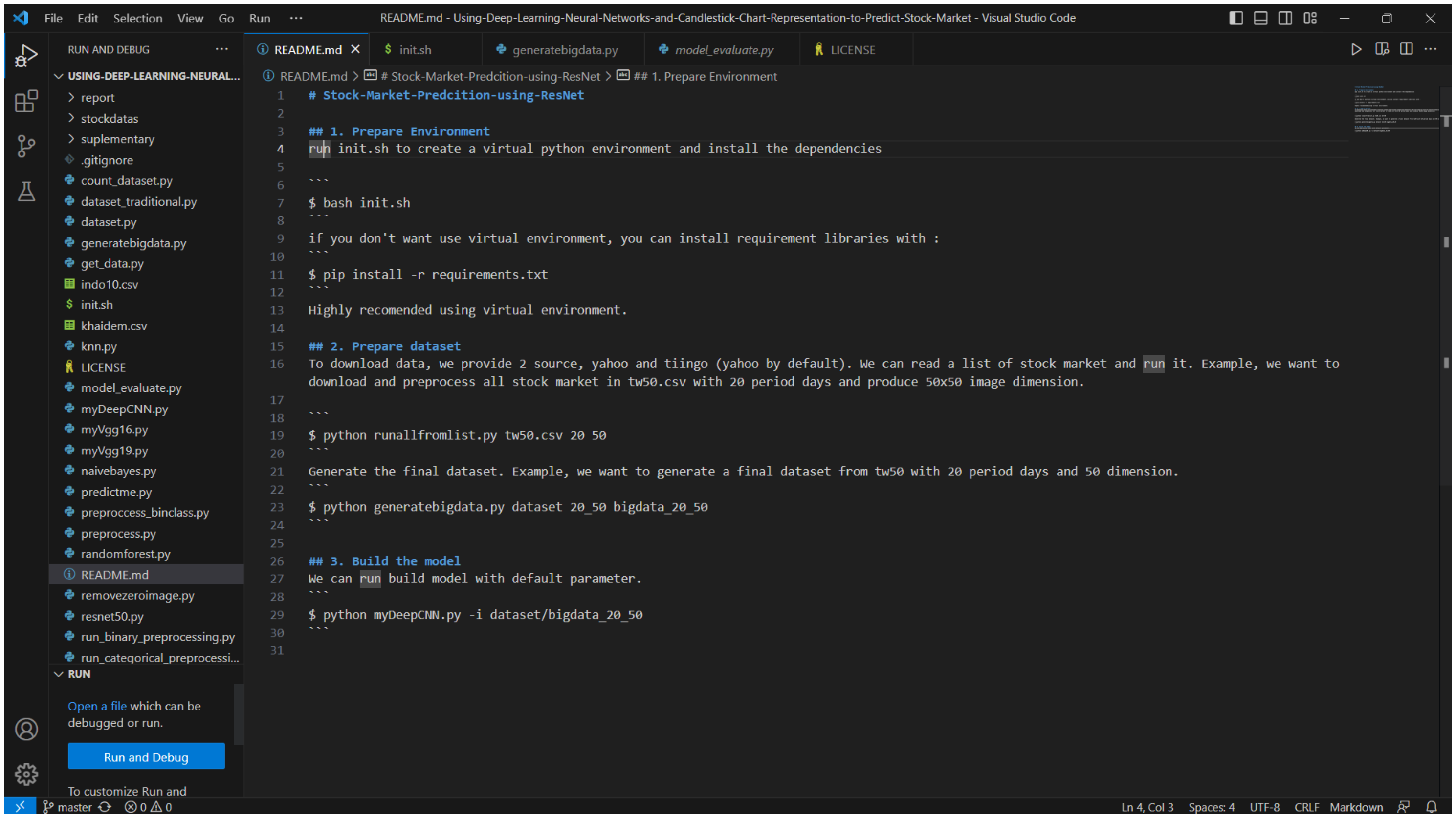
It takes less than 1 minutes to finish the training process and give the output

From the data from Yahoo Finance, the output is the line chart of actual and predicted stock price of Tesla (TSLA) between 2015 and 2020.

3. Test P2 (Attempt)

3.1 Prepare Environment

To start, I have cloned the project (P2) "Stock-Market-Predcition-using-ResNet" by the user "jason887" from https://github.com/jason887/Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market:



~~~~
# Stock-Market-Predcition-using-ResNet

## 1. Prepare Environment
run init.sh to create a virtual python environment and install the dependencies

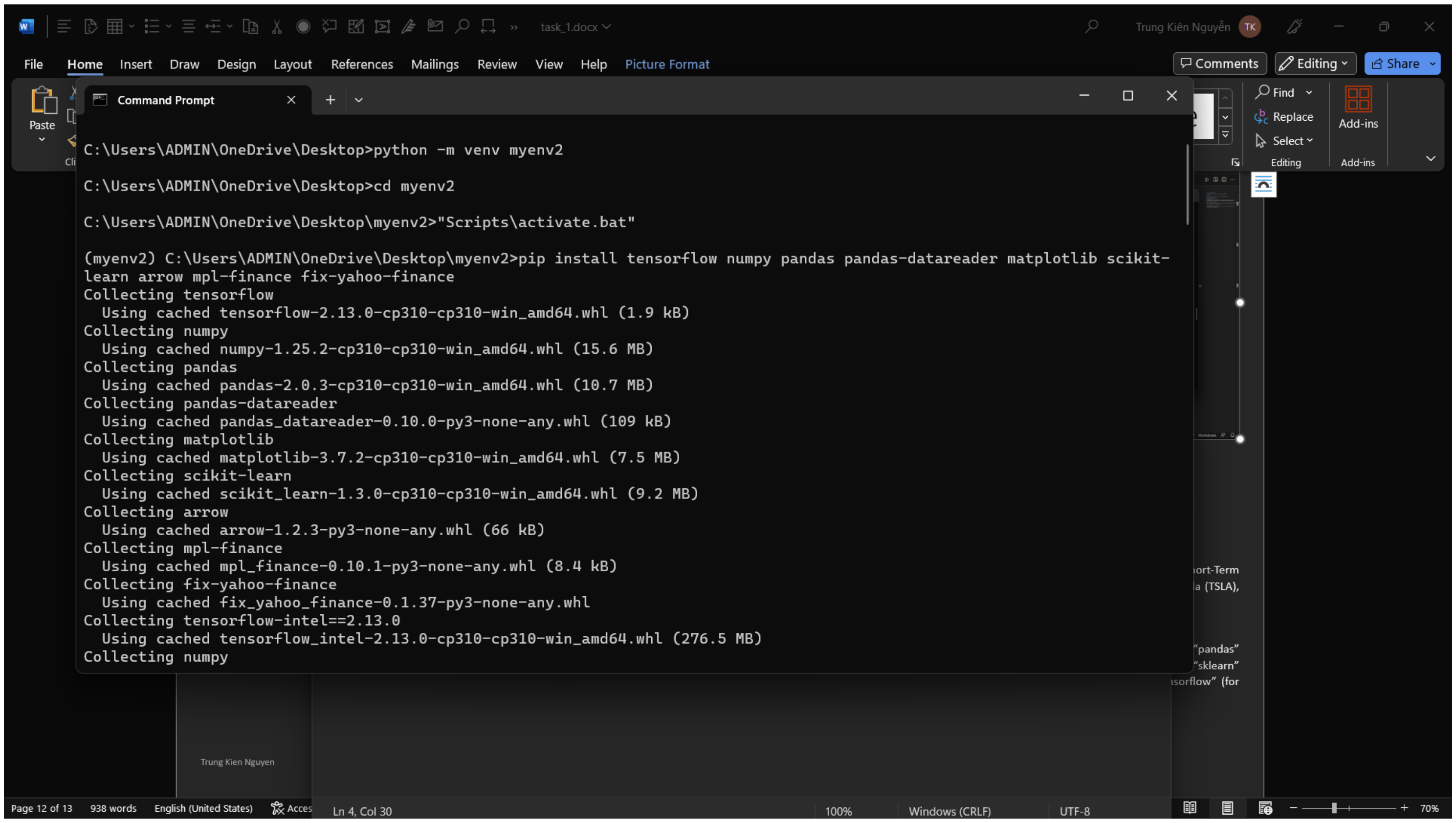
```
$ bash init.sh
```
if you don't want use virtual environment, you can install requirement libraries with :
```
$ pip install -r requirements.txt
```
Highly recomended using virtual environment.

## 2. Prepare dataset
To download data, we provide 2 source, yahoo and tiingo (yahoo by default). We can read a list of stock market and run it. Example, we want to
download and preprocess all stock market in tw50.csv with 20 period days and produce 50x50 image dimension.

```
$ python runallfromlist.py tw50.csv 20 50
```
Generate the final dataset. Example, we want to generate a final dataset from tw50 with 20 period days and 50 dimension.
```
$ python generatebigdata.py dataset 20_50 bigdata_20_50
```


## 3. Build the model
We can run build model with default parameter.
```
$ python myDeepCNN.py -i dataset/bigdata_20_50
```
~~~~

As the setup instructions seem to aim to Linux OS, as well as I can't find the "requirements.txt" file for the project, I have to create a new Python virtual environment directly and install the necessary libraries for the project, instead of trying to run the "init.sh" file:



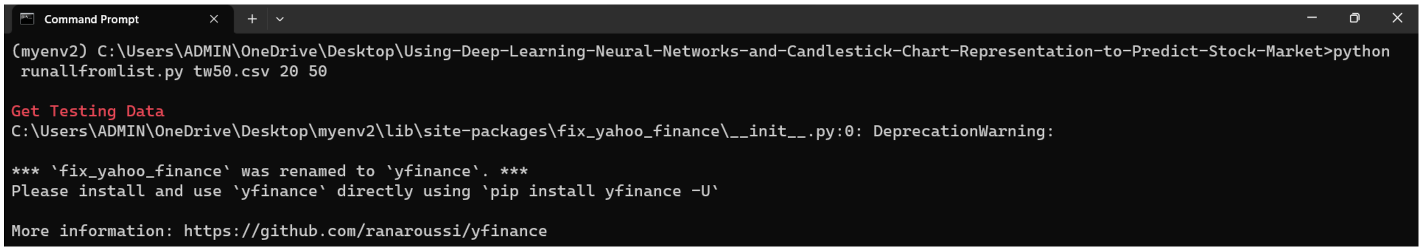
```
C:\Users\ADMIN\OneDrive\Desktop>python -m venv myenv2

C:\Users\ADMIN\OneDrive\Desktop>cd myenv2

C:\Users\ADMIN\OneDrive\Desktop\myenv2>"Scripts\activate.bat"

(myenv2) C:\Users\ADMIN\OneDrive\Desktop\myenv2>pip install tensorflow numpy pandas pandas-datareader matplotlib scikit-
learn arrow mpl-finance fix-yahoo-finance
Collecting tensorflow
  Using cached tensorflow-2.13.0-cp310-cp310-win_amd64.whl (1.9 kB)
Collecting numpy
  Using cached numpy-1.25.2-cp310-cp310-win_amd64.whl (15.6 MB)
Collecting pandas
  Using cached pandas-2.0.3-cp310-cp310-win_amd64.whl (10.7 MB)
Collecting pandas-datareader
  Using cached pandas_datareader-0.10.0-py3-none-any.whl (109 kB)
Collecting matplotlib
  Using cached matplotlib-3.7.2-cp310-cp310-win_amd64.whl (7.5 MB)
Collecting scikit-learn
  Using cached scikit_learn-1.3.0-cp310-cp310-win_amd64.whl (9.2 MB)
Collecting arrow
  Using cached arrow-1.2.3-py3-none-any.whl (66 kB)
Collecting mpl-finance
  Using cached mpl_finance-0.10.1-py3-none-any.whl (8.4 kB)
Collecting fix-yahoo-finance
  Using cached fix_yahoo_finance-0.1.37-py3-none-any.whl
Collecting tensorflow-intel==2.13.0
  Using cached tensorflow_intel-2.13.0-cp310-cp310-win_amd64.whl (276.5 MB)
Collecting numpy
```

Create and setup a new virtual environment of "myenv2" for the project

3.2 Prepare dataset

First, I followed the "README.md" instruction, running the file "runallfromlist.py", downloading and preprocessing all stock market in "tw50.csv" with 20 period days and produce 50x50 image dimension by this command: "python runallfromlist.py tw50.csv 20 50". There are some problems shown up:

- Deprecation Warning

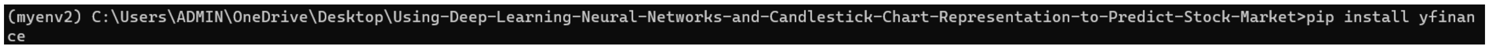


```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>python
 runallfromlist.py tw50.csv 20 50

Get Testing Data
C:\Users\ADMIN\OneDrive\Desktop\myenv2\lib\site-packages\fix_yahoo_finance\__init__.py:0: DeprecationWarning:

*** `fix_yahoo_finance` was renamed to `yfinance`. ***
Please install and use `yfinance` directly using `pip install yfinance -U`

More information: https://github.com/ranaroussi/yfinance
```

"fix_yahoo_finance" was renamed to "yfinance"

Fix:

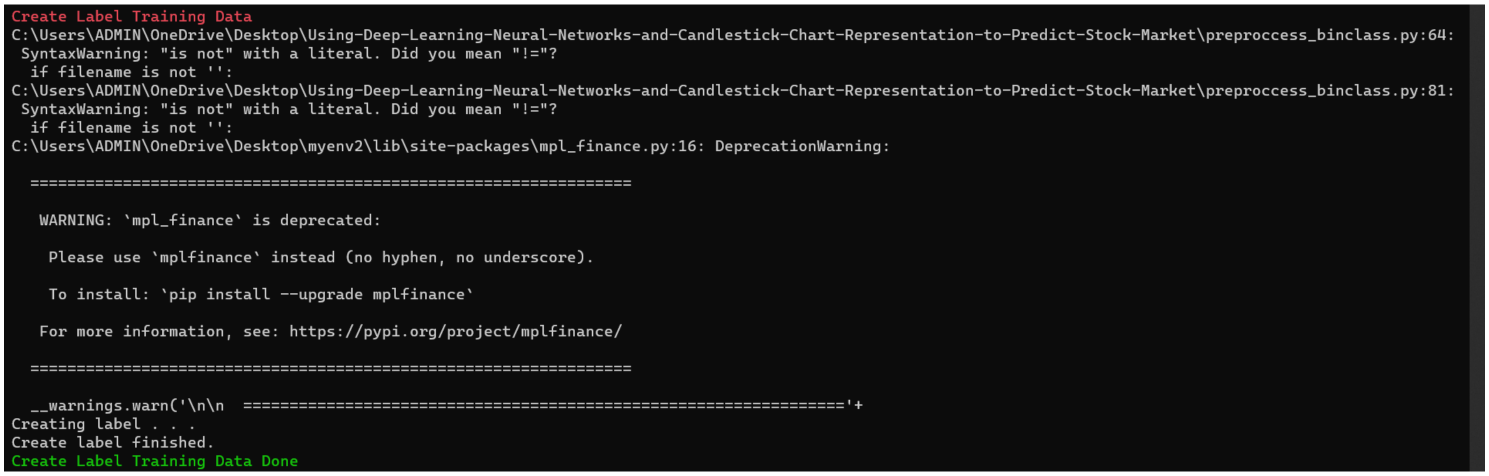
- Install “yfinance”:



```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>pip install yfinan
ce
```

- Change the importing code (from “import fix_yahoo_finance as yf” to “import yfinance as yf”) in the files “get_data.py” and “predictme.py”.

- Deprecation Warning



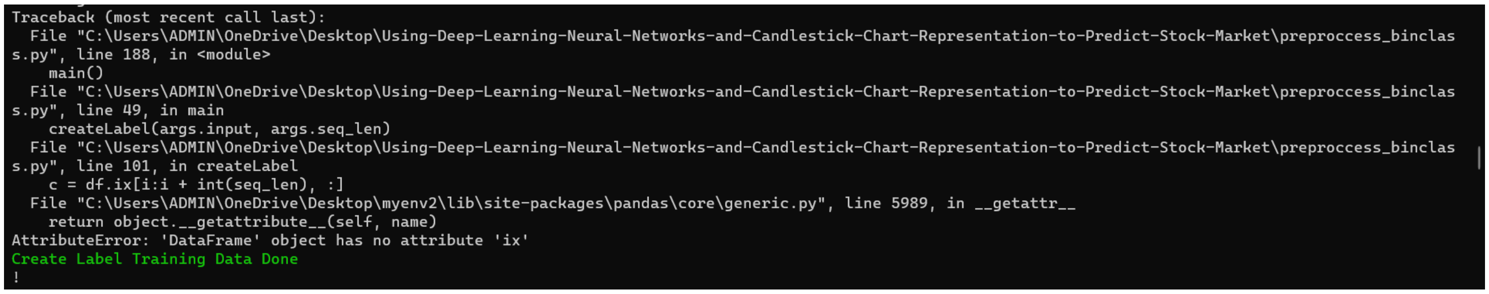
```
Create Label Training Data
C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\preproccess_binclass.py:64:
 SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if filename is not '':
C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\preproccess_binclass.py:81:
 SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if filename is not '':
C:\Users\ADMIN\OneDrive\Desktop\myenv2\lib\site-packages\mpl_finance.py:16: DeprecationWarning:

  =================================================================

   WARNING: `mpl_finance` is deprecated:

    Please use `mplfinance` instead (no hyphen, no underscore).

    To install: `pip install --upgrade mplfinance`

   For more information, see: https://pypi.org/project/mplfinance/

  =================================================================

  __warnings.warn('\n\n  =================================================================='+
Creating label . . .
Create label finished.
Create Label Training Data Done
```

“mpl_financ” is deprecated

Fix:

- Install “mplfinance” using “pip”, then, change the importing code (from “from mpl_finance import …” to “from mplfinance import …”) in the files “predictme.py”, “preprocess_binclass.py”, “preprocess.py”.
  - “preprocess_binclass.py”: from “from mpl_finance import candlestick2_ochl, volume_overlay” to “from mpl_finance import volume_overlay” and “from mpl_finance.original_flavor import candlestick2_ochl”.

- Attribute Error:

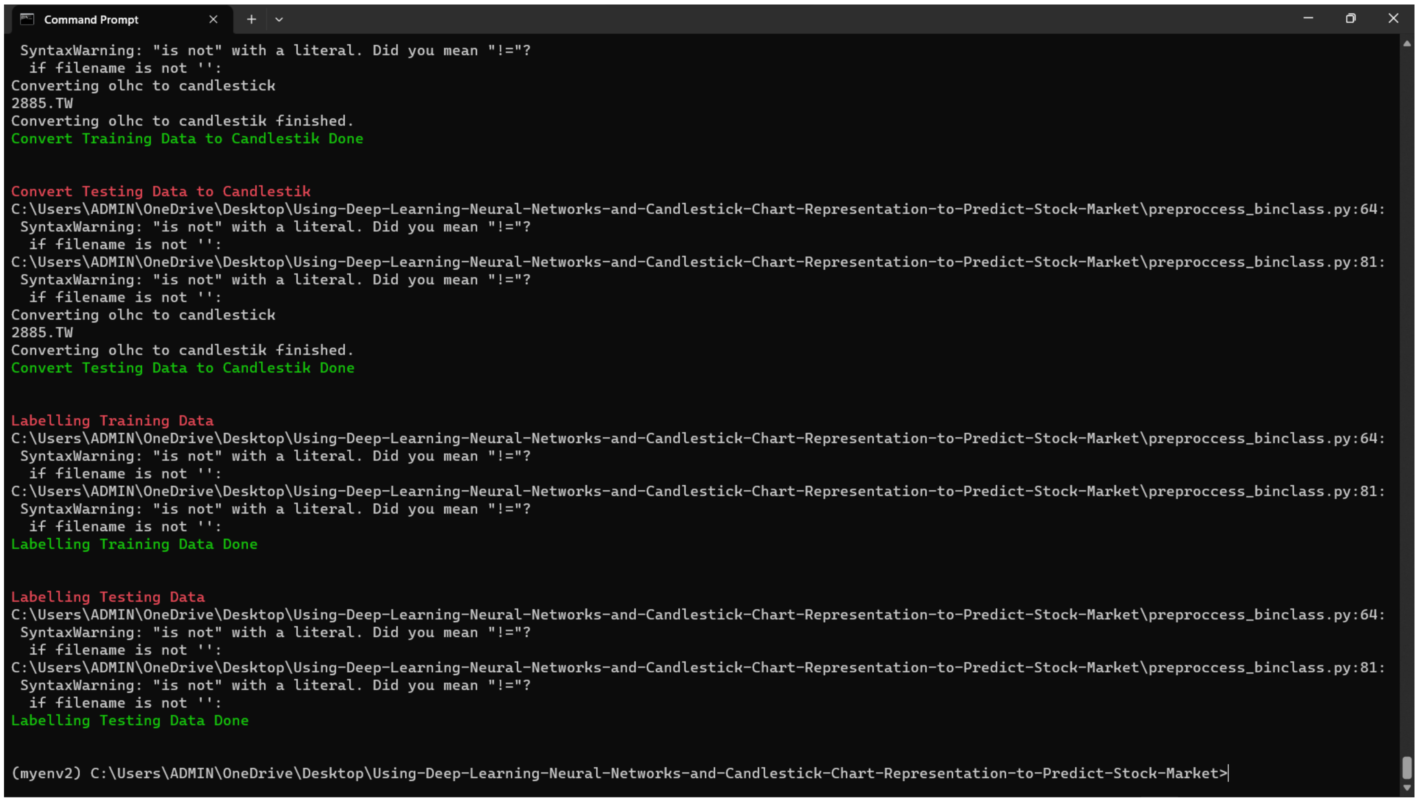


```
Traceback (most recent call last):
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\preproccess_binclas
s.py", line 188, in <module>
    main()
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\preproccess_binclas
s.py", line 49, in main
    createLabel(args.input, args.seq_len)
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\preproccess_binclas
s.py", line 101, in createLabel
    c = df.ix[i:i + int(seq_len), :]
  File "C:\Users\ADMIN\OneDrive\Desktop\myenv2\lib\site-packages\pandas\core\generic.py", line 5989, in __getattr__
    return object.__getattribute__(self, name)
AttributeError: 'DataFrame' object has no attribute 'ix'
Create Label Training Data Done
!
```

“DataFrame” object has no attribute “ix”

Fix:

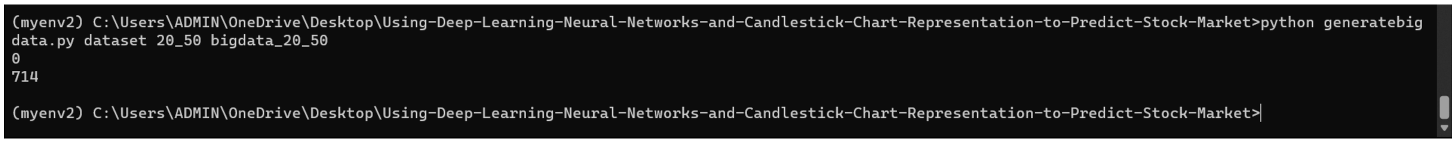
- *In "preprocess_binclass.py" file, line 101 and 147, change from "df.ix[...] to df.loc[...]"*



```
Command Prompt
 SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if filename is not '':
Converting olhc to candlestick
2885.TW
Converting olhc to candlestik finished.
Convert Training Data to Candlestik Done


Convert Testing Data to Candlestik
C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\preproccess_binclass.py:64:
 SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if filename is not '':
C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\preproccess_binclass.py:81:
 SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if filename is not '':
Converting olhc to candlestick
2885.TW
Converting olhc to candlestik finished.
Convert Testing Data to Candlestik Done


Labelling Training Data
C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\preproccess_binclass.py:64:
 SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if filename is not '':
C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\preproccess_binclass.py:81:
 SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if filename is not '':
Labelling Training Data Done


Labelling Testing Data
C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\preproccess_binclass.py:64:
 SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if filename is not '':
C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\preproccess_binclass.py:81:
 SyntaxWarning: "is not" with a literal. Did you mean "!="?
  if filename is not '':
Labelling Testing Data Done

(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>
```

The process now finishes in 9 minutes.

Second, I have generated the final dataset from "tw50" with 20 period days and 50 dimension:

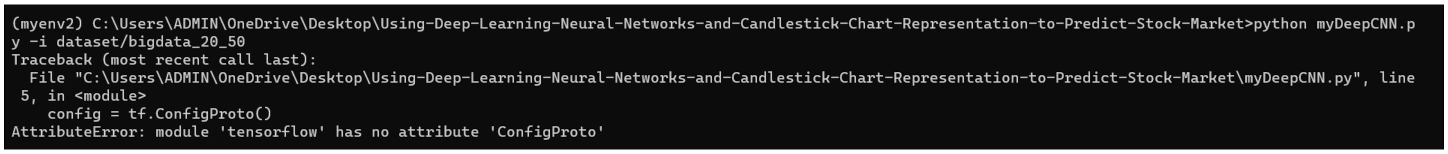


```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>python generatebig
data.py dataset 20_50 bigdata_20_50
0
714

(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>
```

3.3. Build the model

In this stage, I have run "myDeepCNN.py" with default parameters to build the model. There are some problems have shown up:

- Attribute Error

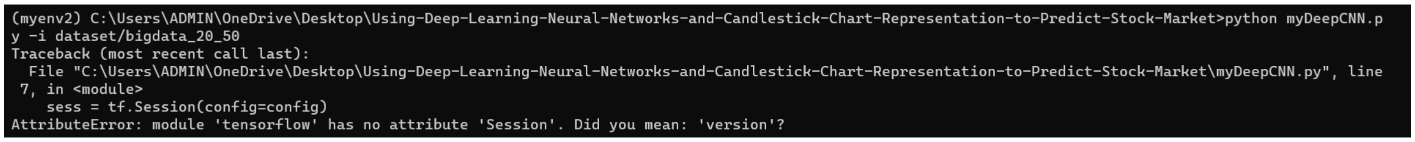


```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>python myDeepCNN.p
y -i dataset/bigdata_20_50
Traceback (most recent call last):
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 5, in <module>
    config = tf.ConfigProto()
AttributeError: module 'tensorflow' has no attribute 'ConfigProto'
```

"tensorflow" has no attribute "ConfigProto"

Fix: “ConfigProto” is disappeared in “tensorflow” 2.0, so I change the code in the file “myDeepCNN.py”, line 5, from “config = tf.ConfigProto()” to “config = tf.compat.v1.ConfigProto()”.

- Attribute Error

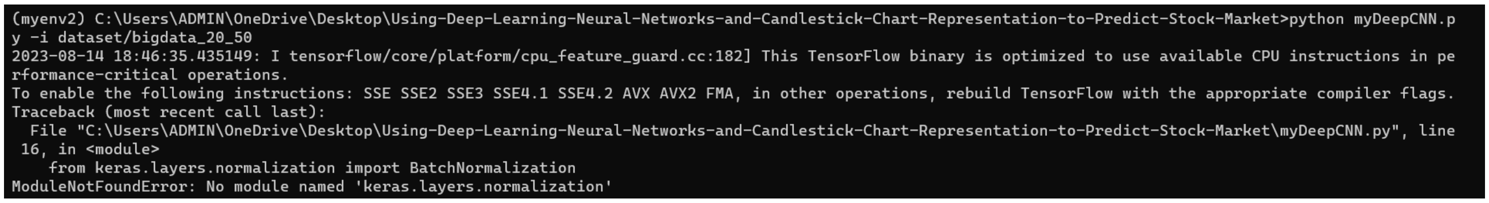


```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>python myDeepCNN.p
y -i dataset/bigdata_20_50
Traceback (most recent call last):
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 7, in <module>
    sess = tf.Session(config=config)
AttributeError: module 'tensorflow' has no attribute 'Session'. Did you mean: 'version'?
```

“tensorflow” has no attribute “Session”

Fix: “Session” is disappeared in “tensorflow” 2.0, so I change the code in the file “myDeepCNN.py”, line 7, from “sess = tf.Session(config=config)” to “sess = tf.compat.v1.Session(config=config)”.

- Module Not Found Error:

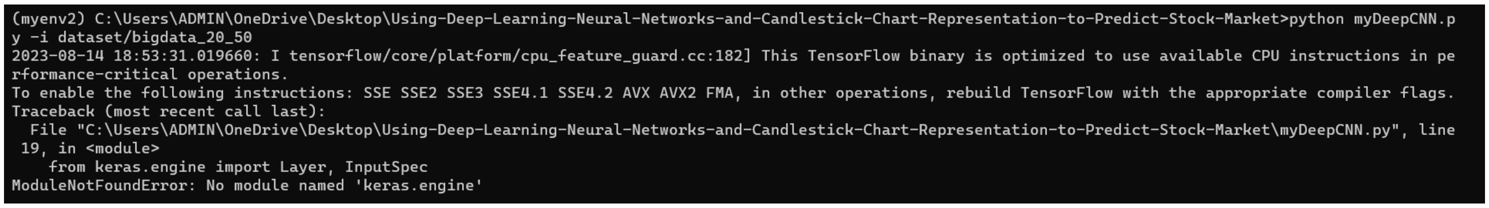


```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>python myDeepCNN.p
y -i dataset/bigdata_20_50
2023-08-14 18:46:35.435149: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in pe
rformance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Traceback (most recent call last):
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 16, in <module>
    from keras.layers.normalization import BatchNormalization
ModuleNotFoundError: No module named 'keras.layers.normalization'
```

No module named “keras.layers.normalization”

Fix: Change the code in the file “myDeepCNN.py”, line 16, from “from keras.layers.normalization import BatchNormalization” to “from keras.layers import BatchNormalization”.

- Module Not Found Error:

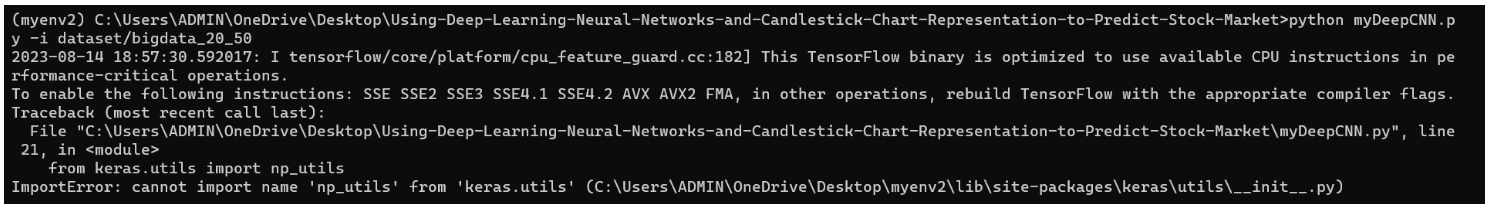


```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>python myDeepCNN.p
y -i dataset/bigdata_20_50
2023-08-14 18:53:31.019660: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in pe
rformance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Traceback (most recent call last):
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 19, in <module>
    from keras.engine import Layer, InputSpec
ModuleNotFoundError: No module named 'keras.engine'
```

No module named “keras.engine”

Fix: Change the code in the file “myDeepCNN.py”, line 19, from “from keras.engine import Layer, InputSpec” to “from keras.layers import Layer, InputSpec”.

- Import Error:

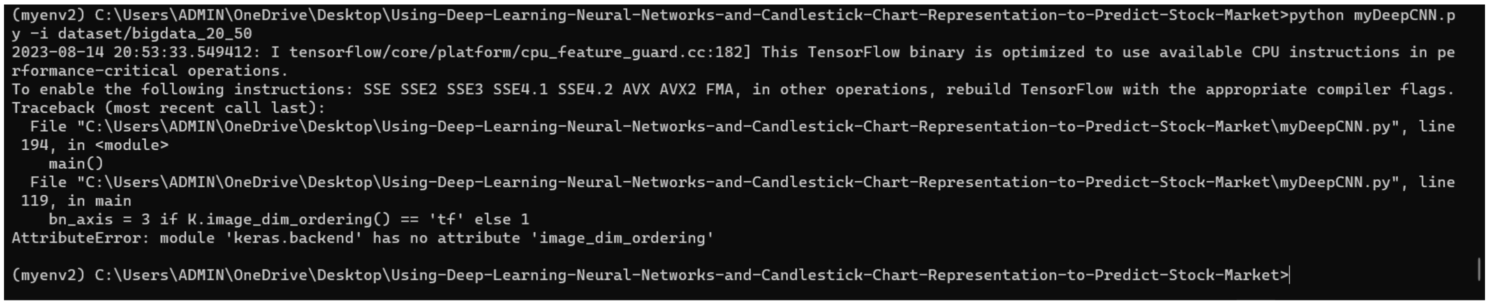


```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>python myDeepCNN.p
y -i dataset/bigdata_20_50
2023-08-14 18:57:30.592017: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in pe
rformance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Traceback (most recent call last):
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 21, in <module>
    from keras.utils import np_utils
ImportError: cannot import name 'np_utils' from 'keras.utils' (C:\Users\ADMIN\OneDrive\Desktop\myenv2\lib\site-packages\keras\utils\__init__.py)
```

Cannot import name "np_utils" from "keras.utils"

Fix: Change the code in the file "myDeepCNN.py", line 21, from "from keras.utils import np_utils" to "from keras.src.utils import np_utils".

- Attribute Error:

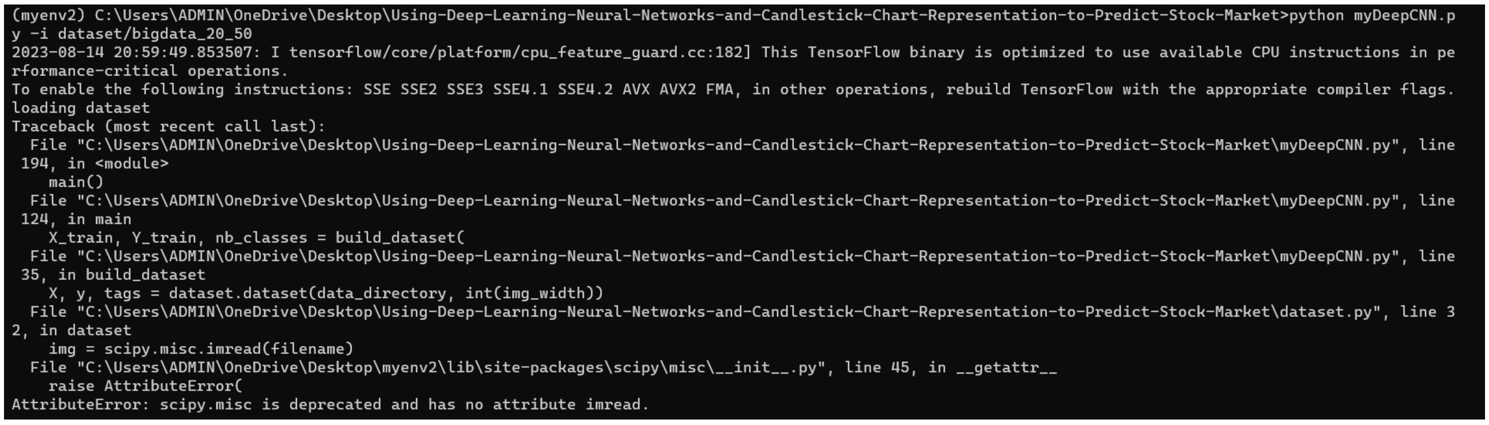


```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>python myDeepCNN.p
y -i dataset/bigdata_20_50
2023-08-14 20:53:33.549412: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in pe
rformance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Traceback (most recent call last):
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 194, in <module>
    main()
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 119, in main
    bn_axis = 3 if K.image_dim_ordering() == 'tf' else 1
AttributeError: module 'keras.backend' has no attribute 'image_dim_ordering'

(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>
```

module "keras.backend" has no attribute "image_dim_ordering"

Fix: In line 119, I have change the code of the file "myDeepCNN.py", from "K.image_dim_ordering()" to "K.image_data_format()".

- Attribute Error:

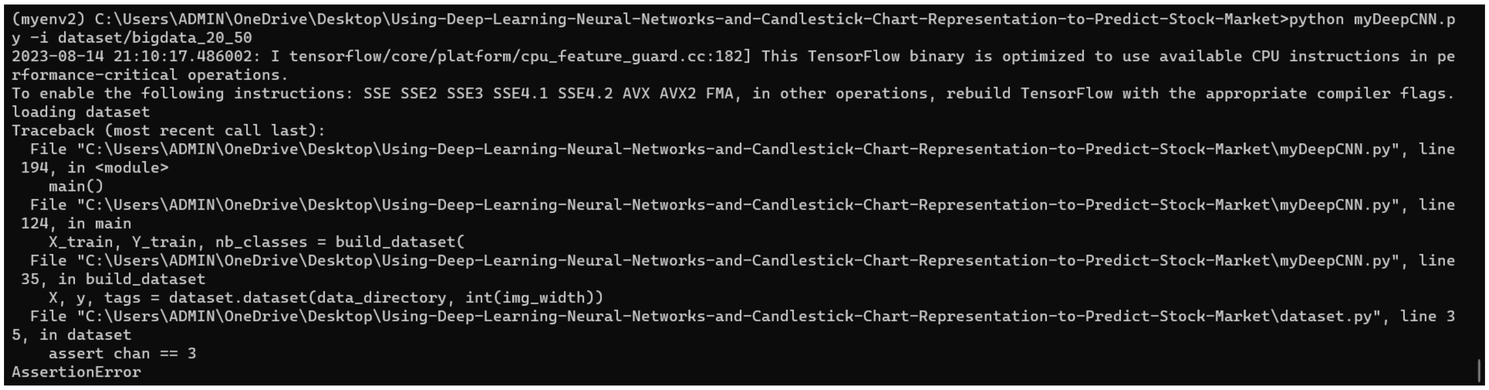


```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>python myDeepCNN.p
y -i dataset/bigdata_20_50
2023-08-14 20:59:49.853507: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in pe
rformance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
loading dataset
Traceback (most recent call last):
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 194, in <module>
    main()
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 124, in main
    X_train, Y_train, nb_classes = build_dataset(
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 35, in build_dataset
    X, y, tags = dataset.dataset(data_directory, int(img_width))
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\dataset.py", line 3
2, in dataset
    img = scipy.misc.imread(filename)
  File "C:\Users\ADMIN\OneDrive\Desktop\myenv2\lib\site-packages\scipy\misc\__init__.py", line 45, in __getattr__
    raise AttributeError(
AttributeError: scipy.misc is deprecated and has no attribute imread.
```

"scipy.misc" is deprecated and has no attribute "imread"

Fix: Import the library "imageio" using "pip", then in the file "myDeepCNN.py", line 5, change the code from "import scipy.misc" to "import image.io", and line 33, from "scipy.misc.imread" to "imageio.imread".

- Assertion Error:

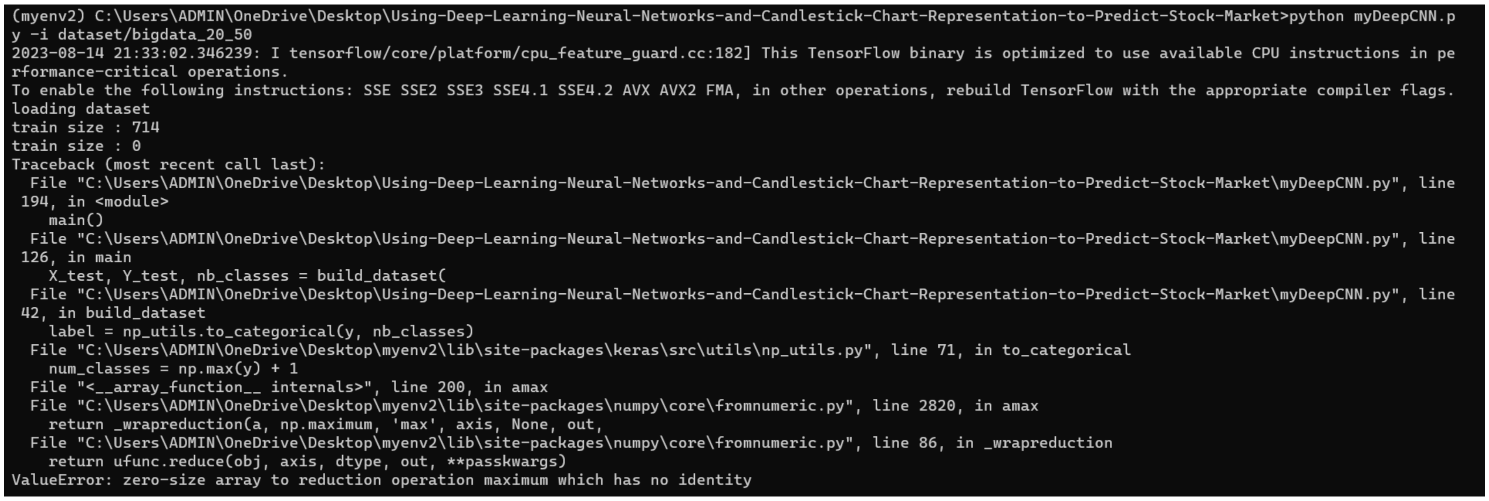


```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>python myDeepCNN.p
y -i dataset/bigdata_20_50
2023-08-14 21:10:17.486002: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in pe
rformance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
loading dataset
Traceback (most recent call last):
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 194, in <module>
    main()
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 124, in main
    X_train, Y_train, nb_classes = build_dataset(
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 35, in build_dataset
    X, y, tags = dataset.dataset(data_directory, int(img_width))
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\dataset.py", line 3
5, in dataset
    assert chan == 3
AssertionError
```

"scipy.misc" is deprecated and has no attribute "imread"

Fix: In the file "dataset.py", delete the line 35 to ignore issues like Grayscale Images or Incorrect Image Formats.

- Value Error **(Unsolved)**



```
(myenv2) C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>python myDeepCNN.p
y -i dataset/bigdata_20_50
2023-08-14 21:33:02.346239: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in pe
rformance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
loading dataset
train size : 714
train size : 0
Traceback (most recent call last):
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 194, in <module>
    main()
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 126, in main
    X_test, Y_test, nb_classes = build_dataset(
  File "C:\Users\ADMIN\OneDrive\Desktop\Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market\myDeepCNN.py", line
 42, in build_dataset
    label = np_utils.to_categorical(y, nb_classes)
  File "C:\Users\ADMIN\OneDrive\Desktop\myenv2\lib\site-packages\keras\src\utils\np_utils.py", line 71, in to_categorical
    num_classes = np.max(y) + 1
  File "<__array_function__ internals>", line 200, in amax
  File "C:\Users\ADMIN\OneDrive\Desktop\myenv2\lib\site-packages\numpy\core\fromnumeric.py", line 2820, in amax
    return _wrapreduction(a, np.maximum, 'max', axis, None, out,
  File "C:\Users\ADMIN\OneDrive\Desktop\myenv2\lib\site-packages\numpy\core\fromnumeric.py", line 86, in _wrapreduction
    return ufunc.reduce(obj, axis, dtype, out, **passkwargs)
ValueError: zero-size array to reduction operation maximum which has no identity
```

Zero-size array to reduction operation maximum which has no identity

SUMMARY

The provided code of **v0.1** aims to build a kind of prediction model using LSTM (Long Short-Term Memory) neural networks, predicting the stock price of a given company, which is Tesla (TSLA), using historical data. These are the summary of its main functionality:

- *Importing necessary libraries*: These include “numpy” (for numerical operations), “pandas” (for manipulating data), “pandas-datareader” (for retrieving financial data), “sklearn” (“scikit-learn”) (for preprocessing data), “matplotlib” (for visualization), “tensorflow” (for building neural networks), and “yfinance” (for downloading stock data)

- *Load data:* Historical stock price for the company Tesla is downloaded from Yahoo Finance using “yfinance” library. The data is loaded within a specific date range between 1/1/2015 and 1/1/2020, and only the "Close" prices are considered.

- *Prepare data:* The loaded “Close” stock price data is normalized using “MinMaxScaler”, a class of “sklearn” library, to transfer them into the range of 0 to 1. Then, it constructs training data samples using a sliding window approach. Each training sample consists of the closing prices of the past.

- *Build the Model:* A sequential LSTM model is created using the Keras, a subpackage of the library “tensorflow”. The model architecture includes multiple LSTM layers for capturing temporal patterns in the data. Dropout layers are added to mitigate overfitting. The model is set up to predict the next day's closing price.

- *Training the Model:* The model is compiled with the Adam optimizer and the mean squared error loss function. After that, it's trained on the constructed training data (“x_train” and “y_train”) using a specified number of epochs and batch size.

- *Test the Model:* The model's performance is evaluated on a separate test dataset, which is downloaded in a similar manner as the training data. The script constructs inputs for the model from the historical data, makes predictions, and scales the predictions back to their original range.

- *Plot Predictions:* The script uses "matplotlib" to plot the actual and predicted stock prices on the same graph. The x-axis represents time, and the y-axis represents stock prices. Actual prices are plotted in black, while predicted prices are plotted in green.

- *Predict Next Day:* The script attempts to predict the stock price for the next day using the most recent "PREDICTION_DAYS" data points. It scales the data, feeds it into the trained model, and then inverts the scaling to obtain the final prediction.