Readme File

Contents

[1. Installing Required Tools 3](#_Toc152418285)

[4. Configuring the Tools 4](#_Toc152418286)

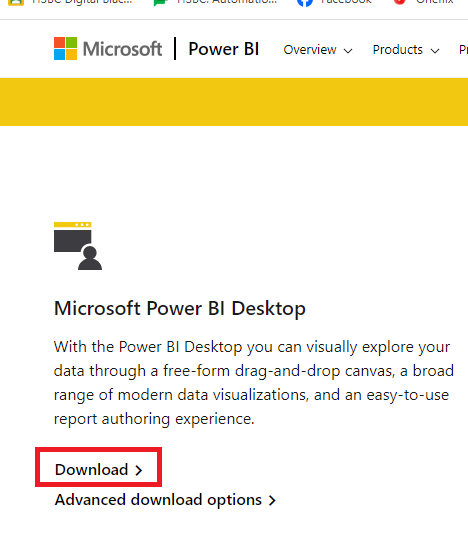
[6. Source Codes 5](#_Toc152418287)

[4. Data Sources 6](#_Toc152418288)

[5. Running the codes 6](#_Toc152418289)

# Installing Required Tools

**Power BI Desktop**

1. Go to this page - <https://powerbi.microsoft.com/en-us/downloads/> and click on Download under Microsoft Power BI Desktop section.
2. 
3. Run the installation file to setup Power BI Desktop on your PC/Laptop.

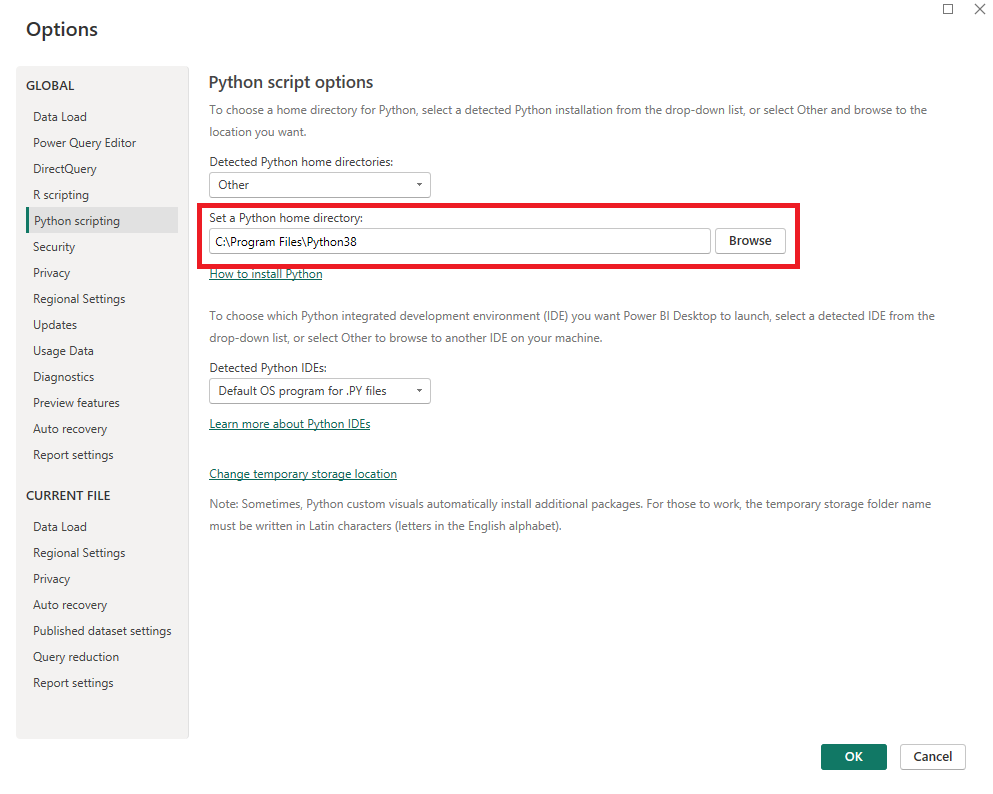
**Python for Windows**

1. Goto this page - <https://www.python.org/downloads/> and download the latest version of PythonA screenshot of a computer

   Description automatically generated
2. Run the installation file to setup Python on your PC/Laptop.

# Configuring the Tools

1. Open Power BI Desktop and go to File>Option & Settings>Option
2. Go to Python Scripting and Set a Python home directory. This is the path where you have installed your python. You can click on Browse to locate the directory.



1. Following Python libraries are used in our code. Please make sure all the libraries are installed
   * math
   * yfinance
   * numpy
   * pandas
   * sklearn
   * matplotlib
   * tensorflow
   * arch
   * datetime
2. For example, to install yfinance, go to your command prompt and run the following command

pip install yfinance

1. To run the python code outside Power BI, we have used jupyter notebook for python. In order to install the Jupyter Notebook, please run the below command in your command prompt

pip install notebook

# Source Codes

In our project, the final outcome is the Power BI Dashboard that includes the python codes to run the model along with data management and visualizations to display visuals to the end users. Before running the python codes in Power BI, we ran the code seperately in Python to make sure the libraries and model are working properly. It is important to note that the python codes required slight modifications to run it in Power BI mainly because we need to store the data in Power BI dataframe for visualization purpose. We also had to join/append Power BI dataframes post running the code to finalise the source for visualization. Please see the explanation of the codes that we are including for this project.

* **Model- LSTM Combined with GARCH DCC Model.ipynb**

This is the python jupyter notebook file that we ran individually in Python outside Power BI. In our project, we have worked with 10 stocks, however this code contains only 1 stock (NVDA). The same code can be used to run for other 9 stocks by just changing the stock code in the following line

stock\_data = yf.download('NVDA', start='2018-11-26', end='2023-11-24')

* **StockInfo.txt**

This file contains the Power Query script for one of the Power BI dataframes. Basically this code reads data from an excel file (one of the data sources used) and uses logics to set different commentaries/explnation for each of the stocks based on the values read from the source file for each stocks.

* **NVDA.txt**

This file contains the Power Query script for one of the Power BI dataframes. In this code, we are using python code to fetch data for NVDA stock from yfinance using yfinance library for last 5 years. Post download, we are running Garch LSTM hybrid model on the data to predict the stock price and finally we are storing the actual and predicted price in a dataframe for NVDA stock.

Important note: We have the similar dataframe and code for all 10 stocks. We are just including one example here to avoid redundancy. In the final Power BI file (FL\_Dashboard.pbix), you can see all the codes in the data section.

* **Stock\_Combined.txt**

This file contains the Power Query script for one of the Power BI dataframes. In this code, we are just combining the dataframes for all 10 stocks into a final dataframe.

* **FL\_Dashboard.pbix**

This is the final dashboard file that contains all the codes and visualization.

A white screen with colorful text

Description automatically generated

# 4. Data Sources

* **StockInfo.xlsx**

This data source file contains all the fundamental financial parameters of all 10 stocks which has been pulled manually from Yahoo Finance page populated into the file. Initially we intended to use yfinance api to pull the same information to avoid using any file based data source. However, due to the server issue at yahoo finance the API did not work and we have chosen to follow this approach.

* 5 years stock data are being pulled from Yahoo Finance using yfinance python library, therefore there is no file based source for this activity.

# 5. Running the codes

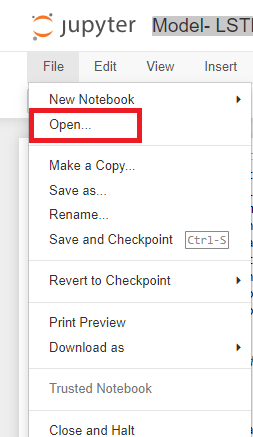
As explained above, we ran the python code individually in Jupyter Notebook first to ensure the model is working properly. This section will describe how to run the code individually in Jupyter notebook as well as in Power BI.

**Running the code in Jupyter Notebook**

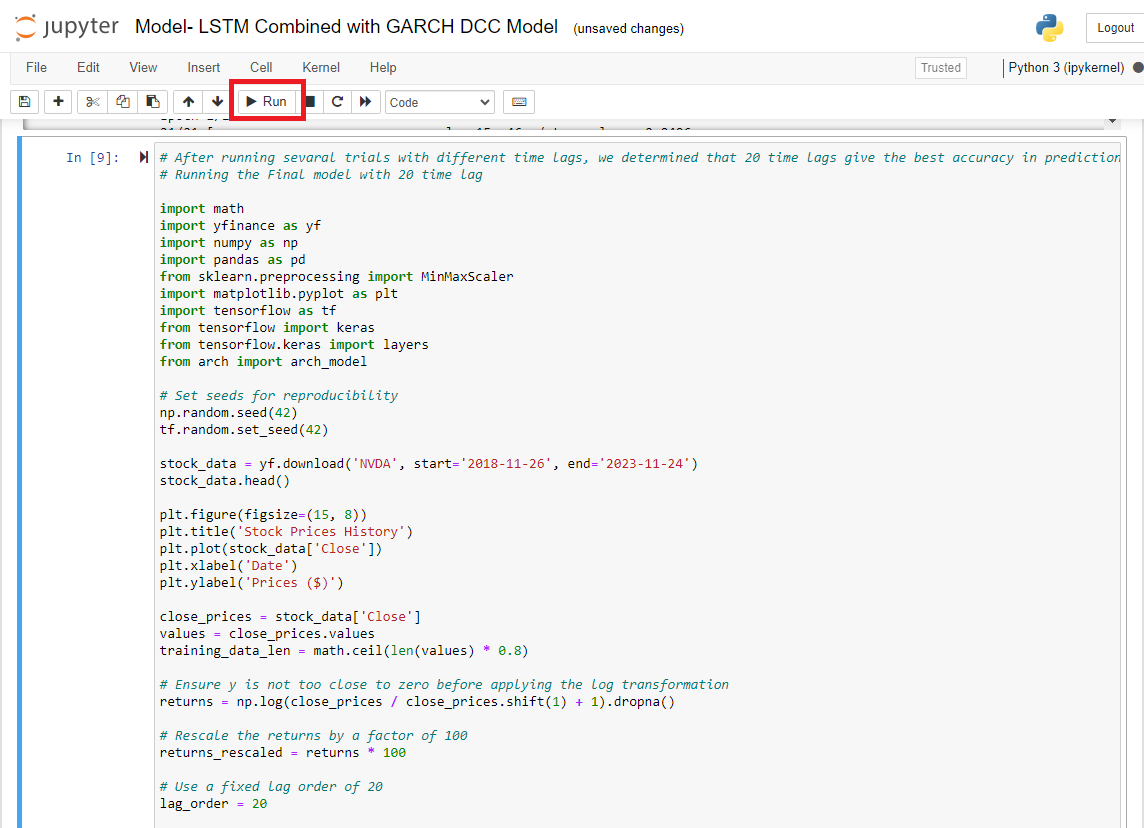
1. To open Jupyter notebook in your PC/Laptop by running the following command in command prompt.

jupyter notebook

1. This will open the Jupyter notebook in your PC/Laptop using your default browser.
2. Once opened, Go to File>Open to locate the file **Model- LSTM Combined with GARCH DCC Model.ipynb** that you have saved in your PC/Laptop



1. You can run each code chunks from Line 9 as the codes above Line 9 were the trial runs to determine the time lags



1. Screenshots of the results below

A graph showing the price of a stock price

Description automatically generated

A number and percentages on a white background

Description automatically generated



**Running the code in Power BI**

1. Create a folder named **temp** in your C drive if not already exist
2. Copy the file **StockInfo.xlsx** from the data source folder and paste in the **temp** folder
3. Double click the **FL\_Dashboard.pbix** to open with Power BI Application
4. Note that you will be able to run the dashboard even without performing the above steps and even without Python installed in your PC as the pbix file already contains all the data after running the model. However, if you want to refresh the data with latest data, you must install python and configure the settings as described in this document
5. Screenshots of the dashboard below

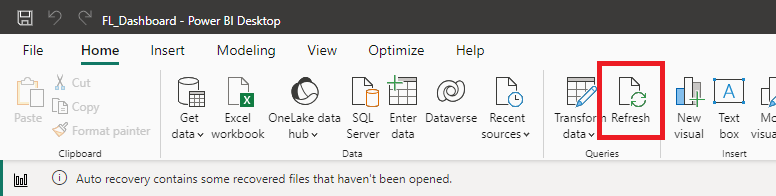
A screenshot of a computer screen

Description automatically generated

A screen shot of a graph

Description automatically generated

1. To refresh the latest data, click on the Refresh button in Power BI



1. Important Note: This may take 1 to 2 hours to refresh all the data as the data is pulled directly from yahoo finance and the model is run on each 10 stocks.