

BCS QuantVision Resources

Week 0:

- **Google Colab-**
📺 Get started with Google Colaboratory (Coding TensorFlow)
- **Github Intro-**
📺 A brief introduction to Git for beginners | GitHub (watch till Basic Git concepts only)
📺 Git and GitHub Tutorials #4 - Creating Forks and Pull Requests
- **Python Basics-** <https://www.w3schools.com/python/> (just get a brief idea of applications and syntax, no need to bore yourself to death memorizing them all)
- **Numpy-** <https://www.w3schools.com/python/numpy/default.asp> (till array filter) .
- **Pandas:** <https://www.w3schools.com/python/pandas/default.asp> (till pandas analyzing data).
- **Matplotlib:** https://www.w3schools.com/python/matplotlib_intro.asp (till pie charts)

Go through these notebooks for a better idea of above concepts:

🔗 BCS QV Python Basics.ipynb

🔗 BCS QV Application.ipynb

DEADLINE to complete these resources : 3rd Dec EOD

Assignments Week 0

1. 🔗 QV_Pandas.ipynb (use Iris dataset sent on group)
2. 🔗 QV_Matplotlib.ipynb
3. 🔗 QV_Numpy .ipynb

DEADLINE to submit Assignments : 6th Dec EOD

Week 1:

- **Understanding with candlestick patterns**

 Candlestick charts: The ULTIMATE beginners guide to reading a cand...

Some investopedia articles to familiarise yourself with common patterns:

1. <https://www.investopedia.com/trading/candlestick-charting-what-is-it/>
2. <https://www.investopedia.com/articles/active-trading/062315/using-bullish-candlestick-patterns-buy-stocks.asp>
3. <https://www.investopedia.com/terms/b/bearishengulfingp.asp>

- **Data Collection Resources**

We'll be using financial websites to collect relevant data, therefore it is essential for you to know how to use them

1. **yfinance (Yahoo Finance)** - Best for stocks
<https://www.geeksforgeeks.org/python/how-to-use-yfinance-api-with-python/>
2. **Alpha Vantage** - Renders comprehensive financial data
<https://algotrading101.com/learn/alpha-vantage-guide/>

- **Familiarise yourself with tickers**

<https://www.investopedia.com/articles/01/070401.asp>

Assignment Week 1

Deliverable: A Google Colab Notebook (.ipynb).

1. In a code cell, install libraries like yfinance, plotly (for interactive plotting)

2. Select a stock (any ticker of your choice) and use yfinance to download historical data for this stock. Get the last 6 months of data and set the interval to 1 day.
3. Now, visualise the numbers. Write code to create an interactive chart:
 - Use plotly.graph_objects to create a Candlestick chart.
 - Map the dataframe columns (Open, High, Low, Close) to the corresponding parameters in the chart function.
 - Add a title to the chart
 - **Requirement:** The chart must be interactive (meaning you can zoom in and out to see individual candles clearly).
4. Analyse the chart you generated and try to look for a pattern. Try to locate one instance of a specific candle pattern mentioned in your reading (e.g., A long bullish candle, a Doji, or a Bearish Engulfing pattern).
5. In a text cell below the chart, write down:
 - The Date the pattern occurred.
 - The Name of the pattern you think it is.
 - Did the price go Up or Down immediately after that pattern?

Week 2:

- Learn what are **Regression Problems, Classification Problems** and how they differ from each other fundamentally.

<https://www.javatpoint.com/regression-vs-classification-in-machine-learning>

- **Linear and Logistic Regression** (and the math behind them- **Gradient Descent**) — beautifully explained in these blogs.

<https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86>

<https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>

- For youtube peeps:

[The OG Andrew NG Playlist\(watch Lectures 2 and 3\)](#)

[Statquest\(Gives nice intuition on evaluation metrics and other stuff\)](#)-Will recommend going through it after completing the previous resources.

- If time permits:(Go through **KNN Algorithm**)

[KNN](#)

- Learn about how to use them using **scikit-learn**.(Think how can you use it for implementing linear and logistic regression)

[Implementation of KNN using scikit-learn](#)

- Gives a very clear intuition on **Neural Networks**.
[The OG 3Blue1Brown Playlist\(Watch first 4 videos\)](#)
- Read about **Perceptrons**
[perceptrons](#)
- Read these articles for further understanding
[this](#) and [this](#)
- To learn about the math of neural networks, read [this](#)

Week 3:

- **Data Augmentation:**
 1. [A Complete Guide to Data Augmentation | DataCamp](#)
 2. [What is Data Augmentation? How Does Data Augmentation Work for Images? - GeeksforGeeks](#) (read about image processing techniques in this if time permits(optional))
- **Tensorflow-** [Introduction to TensorFlow](#) (intro)
[TensorFlow 2 quickstart for beginners bookmark_border](#) (application)

- **PyTorch-** [Learn the Basics — PyTorch Tutorials 2.9.0+cu128 documentation](#) (intro)
[Quickstart — PyTorch Tutorials 2.9.0+cu128 documentation](#) (application)
- **Scikitlearn-**[Learning Model Building in Scikit-learn - GeeksforGeeks](#)
- **Keras-**[Introduction to Keras for engineers](#)
[Keras: The high-level API for TensorFlow](#)
[Introduction to Keras for engineers](#) (application)
- **CNN:** A good article on CNNs...make sure to go through it.
[An Introduction to Convolutional Neural Networks: A Comprehensive Guide to CNNs in Deep Learning | DataCamp](#)

[Python Convolutional Neural Networks \(CNN\) with TensorFlow Tutorial | DataCamp](#)(only go through the step by step implementation part)

Assignment 3:

Your goal is to build a CNN model that takes a chart image as input and predicts if the market direction is UP or DOWN. You'll be using the following **dataset**: [Candlestick Image Data](#)
(You can either use the Kaggle notebook and import the dataset or download the dataset on your device and use it)

Tasks:

1. **Data Augmentation** : Use all the data augmentation techniques which you think would be **relevant** in improving the model training.
(as a hint: vertical flip would not be helpful think why)
2. **Build a CNN** :
 - Remember to resize the input images. Use appropriate activation functions. Experiment with the hidden layers(number, activation functions), dropout, pooling etc.

- Try to apply the techniques given in the resources
3. **Results:** Try to make plots with your results(example Training vs Validation Accuracy over time)

Deadline: 3rd EOD