Case Study - Brief

Congratulations! You have been shortlisted for the position of Data Analyst Intern at CogniTensor.
In order to proceed with your application, you need to submit your solution for this case study within **3** days (from the date of receiving this).

Introduction

You will be given an objective, which you need to achieve, or get closer to achieving within the context of Data Science.
The programming language to be used is Python (3.6+), make sure you stick to Python and it's open source libraries.

What you'll receive -

- Input Dataset
- Guidelines and Objectives

What we expect to receive from you -

- Documentation (Should contain the following)
    - Approach (Keep it as detailed as possible)
    - Findings
    - Challenges and Opinions
    - Conclusion
    - Retrospective (What could have been done better)

- Code (Should fit the following specifications)
    - To be sent as a .zip file containing modules properly arranged and pathed according to usage
    - All code should be modular and production friendly
    - Try to write functions and/or classes wherever applicable
    - Unnecessarily iterative code would be penalised
    - Should follow the PEP8 convention and be properly linted (More about the PEP8 convention here)
    - Avoid using Jupyter notebooks, but in case you choose to use them, convert all notebook code to .py files before sending. **.ipynb** files will not be accepted

Treat this case study as a task assigned to you while working on a live project. We will be evaluating your technical skills as well as how effectively you are able to document and communicate your approach to a problem.

CS - 1 : Quantitative Analysis and Modeling for S&P 500

Overview

The meteoric increase in compute power and advances in Machine Learning have given rise to a variety of use-cases for mechanical/algorithmic trading. Quantitative funds across the world use a plethora of techniques to forecast market prices, volumes and general market behaviour.

S&P 500 is one of the world's leading benchmark indices consisting of 500 publicly listed companies. Your study will be restricted to data from these companies' price-volume data (as traded on the New York Stock Exchange)

A detailed data description will be provided further on in this document.

Objective

There are 3 main objectives -

1. **Volatility Index** - Out of all the 500 stocks in the dataset, establish a weekly volatility index which ranks stocks on the basis of intraday price movements.
   (Weekly volatility Index implies that it is to be calculated on a weekly time frame and both intraday as well as weekly change in price needs to be used in calculating volatility)

   a. Give an exploratory analysis on any one stock describing it's key statistical tendencies.
   b. The index should rank the stocks from most to least volatile in the selected time frame.
   c. The output needs to be grouped weekly showing the Top 10 Most and Least Volatile stocks. Both your code and output will be evaluated.

2. **Pair Trading** - The concept of pair trading suggests that there are stocks whose prices move together (could have an inverse relationship). More information on pair trading can be found at https://zerodha.com/varsity/chapter/pair-trading-basics/

   Your objective is to identify the 5 strongest pairs for every year in the dataset (eg. 5 strongest pairs for 2014, 2015 and so on)

3. **Binary Classification -** Given a stock and it's data, you have to predict whether it will close lower than it opened (red) or higher than it opened (green) [Continued on the next page]

   You need to submit your model whose performance will be tested on our test data. (Will be a subset of the data provided to you).
   Your prediction function needs to be standardised to ensure it's compatibility with our test function and should follow the following guidelines -
   Input Arguments - Ticker Symbol, date (to predict for), Historical Price Series for the selected stock (up till the mentioned date, but be sure to avoid look ahead bias)
   Function Returns - **1** (for Green), **0** (for Red), **0.5** (For No Confidence)
   (A 'No Confidence' will be treated as a random prediction and is better than a wrong prediction)

Note -

- We strongly encourage engineering additional features. To give an example, traders usually look at candlestick charts and their activity influences the prices. So you could engineer features to emulate the characteristics of a candlestick eg. Body Size, Upper Shadow, Lower Shadow etc) which can be easily extracted from the price O,H,L,C data.

- If you are engineering additional features, make sure your function extracts them in real time before predicting (We are passing the historical price series as an argument to the function for this purpose itself)

Data Description

The given data has been downloaded from Kaggle and is very clean.
The dataset can be downloaded here and contains the following columns -

- **Date** - The day when trading took place. Please note that if predicting for 10-01-2019, you will not have the data up till 09-01-2019 only. Make sure look-ahead bias is avoided in all your analysis and activities)
- **Open -** Opening price
- **High -** Highest price level reached during the day
- **Low -** Lowest price level reached during the day
- **Close -** Closing price
- **Volume -** Number of stocks traded on that day
- **Name -** Name or Ticker Symbol of the stock

Make sure you follow all the guidelines before beginning with the case study. We wish you good luck and look forward to receiving your solutions!