TASK

1. TASK-1

a. Please create a Chatbot using the RASA framework, the chatbot should be able to do the following tasks.
   i. **The current COVID-19 cases distribution.**
   ii. **Give the official numbers of the current COVID cases in your state (assuming INDIAN states only)**
   iii. **Send COVID updates via Email, if the consumer's request.**
   iv. **Plot the COVID cases on a Map and show it to the requester when asked.**
   v. **Give some general tips about COVID if asked**
   vi. **THIS BOT SHOULD BE CONVERSATIONAL EVEN THO THE INITIAL INTERFACE CAN BE HARDCODED BUT IF A QUERY IS ENTERED BY ANY PERSON IT SHOULD BE ABLE TO GET THE CONTEXT OF THE QUERY AND REVERT BACK WITH A SUITABLE REPLY**

b. You should be able to deploy this chatbot on telegram, messenger, etc if you feel comfortable please deploy it onto some cloud so that we could assess you on your end-to-end deployment skills as well

c. You have **7+2** days to complete this and below mentioned task, the later 2 days extension will only be provided if at least 2 sub-parts are completed and shown **on or before the 7th day**.

d. We don't expect you to use any paid API in getting the information in this task, the goal of this task is to check your Software development skills and end-to-end project deployment skills so even if the data of COVID is not updated we will accept it as a solution as long as the data pipeline is working, you may revert back to us on biz@guidona.com if you require any more information.

e. All the necessary API are basically if you search on Google.

f. Although you are expected to provide us with the working link of the deployed project and the code using which you have made that project if you have some issues with the deployment on the server, we will accept the code only if it is functioning on a local server.

2. TASK-2

a. What you'll receive -
   i. Input Dataset
   ii. Guidelines and Objectives
   iii. What we expect to receive from you -
   iv. Documentation (Should contain the following)
   v. Approach (Keep it as detailed as possible)
   vi. Findings

vii. Challenges and Opinions
viii. Conclusion
ix. Retrospective (What could have been done better)
x. Code (Should fit the following specifications)
xi. To be sent as a .zip file containing modules properly arranged and pathed according to usage
xii. All code should be modular and production-friendly
xiii. Try to write functions and/or classes wherever applicable
xiv. Unnecessarily iterative code would be penalized
xv. Should follow the PEP8 convention and be properly lined (More about the PEP8 convention here)
xvi. 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
xvii. 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.
xviii. Overview

b. The meteoric increase in computational 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 behavior.
c. 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)
d. A detailed data description will be provided further on in this document.
e. OBJECTIVE
    i. 1. Volatility Index
        1. Out of all the 500 stocks in the dataset, establish a weekly volatility index that ranks stocks based on intraday price movements.
        2. (Weekly volatility Index implies that it is to be calculated on a weekly time frame and both intraday, as well as a weekly change in price, needs to be used in calculating volatility)
        3. a. Give an exploratory analysis on any one stock describing its key statistical tendencies.
        4. b. The index should rank the stocks from most to least volatile in the selected time frame.
        5. 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.
    ii. 2. Pair Trading

1. **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/**
2. **a. 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)**

iii. **3. Binary Classification**

1. **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]**
2. **a. 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).**
3. **b. Your prediction function needs to be standardized to ensure it's compatibility with our test function and should follow the following guidelines -**

iv. **Input Arguments**

1. **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)**

v. **Function Returns**

1. **1 (for Green), 0 (for Red), 0.5 (For No Confidence)**
2. **(A 'No Confidence' will be treated as a random prediction and is better than a wrong prediction)**

vi. **Note**

1. **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.**
2. **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)**

vii. **Data Description**

1. **The given data has been downloaded from Kaggle and is very clean.**
2. **The dataset can be downloaded from this link and contains the following columns –**
3. **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)

4. **Open - Opening price**
5. **High - Highest price level reached during the day**
6. **Low - Lowest price level reached during the day**
7. **Close - Closing price**
8. **Volume - Number of stocks traded on that day**
9. **Name - Name or Ticker Symbol of the stock**
10. **Make sure you follow all the guidelines before beginning with the task. We wish you good luck and look forward to receiving your solutions! Try to solve as much as you can and submit before the deadline to be considered for the next step.**

3. Revert back after receiving this task with your **GitHub ID.**
4. **Any delay in the submission will not be tolerated.**