# Data Science Interview Instructions:

#### **Use-case Assignment:**

* **Purpose:** You will receive a data science use-case (either classification or regression) at least 24 hours before your interview.
* **Requirements:**Review the use-case thoroughly and prepare to answer specific questions asked for the use case

#### **Question Answering Process:**

* **Implementation:** Use Python code to answer each question provided for the assigned use-case.
* **Explanation:** Prepare to explain your code and the rationale behind your approach during the interview.

#### **Evaluation Criteria:**

* **Assessment:** Your performance will be evaluated based on:
  + **Code Writing:** Ability to write clean and effective Python code.
  + **Explainability:** Clarity and coherence in explaining your code, choices made, and results obtained.

#### **Development Environment:**

* **IDE Recommendation:** Use a local **Jupyter Notebook or Spyder IDE** for writing your Python code.
* **Markdown Usage:**Utilize Markdown cells in your Jupyter Notebook for formatting and structuring your answers to each question.

#### **Code Readability:**

* **Clean Code:** Ensure your code is well-organized and easily readable.
* **Comments:** Add explanatory comments throughout your code to enhance understanding during the interview.

#### **Interview Preparation:**

* **Notebook Sharing:** You will be expected to share your Jupyter Notebook during the interview call.
* **Presentation:**Make sure your notebook is well-prepared for sharing and review, emphasizing cleanliness and clarity.

#### **Task Execution Timeline:**

* **Deadline:** Complete the assigned tasks and prepare your notebook for the interview within the provided timeframe.
* **Review:** Double-check your work for completeness and correctness before the interview session.

# Data Science Interview Task: Predicting Diabetes

Objective**:**

You are tasked with developing a predictive model to predict the quantitative measure of disease progression one year after baseline by using the dataset from sklearn package. The goal is to build and evaluate a model that accurately predicts the progression by answering certain questions asked.

Dataset**:**

The dataset has 10 baseline variables, age, sex, body mass index, average blood

pressure, and six blood serum measurements were obtained for each of n = 442 diabetes patients, as

well as the response of interest, a quantitative measure of disease progression one year after baseline.

**Note:** You can **load** the diabetes dataset from “**sklearn.dataset“** package using **“load\_wine ()”.**

Questions:

* + Q1: Write the code for below sub-questions
    - Generate last 3 rows of dataset?
    - Print First, second and last row in the Dataframe
    - Print the 2 rows just before the last 3 rows in the Dataframe
    - Summarize the data as an analyst
  + Q2: Subset the above DataFrame for age>35.
  + Q3: Find relation between Age, Sex and Target Variable.
  + Q4: Generate Box plot to detect outliers.
  + Q5: Split the dataset into training and test sets.
  + Q6: Train the Model using training dataset.
  + Q7. Generate Predictions using Test dataset.
  + Q8: Calculate MSE for the predictions.
  + Q9: Can you evaluate your models?
  + Q10: Which model would you suggest?
  + Q11: What would be you model monitoring strategy, which key performance statistical metrics you will consider while looking at model performance?

# Do’s & Don’ts in while writing the code:

Do’s:

1. You can use google to search for python coding.
2. Use matplotlib & seaborn packages for plotting
3. Make the plots clean and clutter free.
4. Adhere to the instructions provided regarding the coding tools & format.
5. Organize your code logically with section headings & comments.
6. Seek clarification if unsure about any aspects of the task or instructions to avoid misunderstandings.
7. Articulate the reasoning behind your code decisions, model & hyper-parameter selection during the interview.
8. Validate your code and models thoroughly before sharing in the interview.

Don’ts:

1. Discard the specific tools, formats or submission guidelines provided for the task.
2. Use google-colab for coding.
3. Submit codes that is poorly structured, lack comments or uses unclear variable names, making it difficult to understand.
4. Use unnecessarily complex solutions that are difficult to follow or maintain.
5. Present code that is copied from external sources without proper under understanding.