

# First attempt tech task

So for this task you will work with the dataset attached to this message in the link provided below, the deadline is in 2 weeks after the moment you received the task:

## Healthcare Diabetes Dataset

A Comprehensive Dataset for Diabetes Risk Assessment

<https://www.kaggle.com/datasets/nanditapore/healthcare-diabetes>



These are some main steps that should be done in your work, but feel free to add anything else that you think is needed:

1. Try to understand the problem that you have to solve and make some assumptions and analysis based on the data that you have
2. According to the task, clean the data and extract the most out of it
3. Create a model, train it on train data and make predictions for test data
4. Check if your initial assumption relates to the results you got and make some conclusions
5. Present your work in a Jupyter notebook with proper comments and markdown paragraphs of your ideas/assumptions/conclusions

Once you are done, upload your work to a GitHub repository and send it to us so we could review it.

A technical interview will be scheduled where you will have to present what you did, answer our questions related to your work and go through a live session of Python Skill check. We are expecting from our interns an Intermediate level of knowledge in Python programming skills so make sure you refresh your knowledge if you feel the need to do so.

Deadline for presenting the Jupyter Notebook: 2 weeks from the day the task was sent.

Topics that an Intermediate Python Programmer should be familiar with:

1. Basic Python syntax: Understanding variables, data types, loops, conditional statements, and functions.
2. Data structures: Lists, tuples, sets, dictionaries, and comprehensions.
3. File handling: Reading and writing text, CSV, and JSON files.
4. Libraries and packages: Importing and using built-in and third-party libraries.
5. Functions and lambda expressions: Creating and using functions, and understanding lambda expressions and their use in functional programming.
6. Object-oriented programming: Classes, objects, inheritance, and other OOP concepts.
7. Error handling: Understanding and handling exceptions using try, except, and finally blocks.
8. Regular expressions: Using regular expressions for pattern matching and text manipulation.
9. Web scraping: Extracting data from websites using libraries like Selenium or BeautifulSoup and requests.
10. APIs: Interacting with APIs to fetch and process data from external (example API - <https://www.geeksforgeeks.org/python-find-current-weather-of-any-city-using-openweathermap-api/>)
11. Basic understanding of Data Structures and Algorithms: Familiarity with fundamental data structures (lists, dictionaries, sets, tuples, stacks, queues, trees, etc.) and algorithms (sorting, searching and others).
12. Linear Algebra: A basic understanding of linear algebra concepts, such as vectors, matrices, and matrix operations.
13. Calculus: A foundational understanding of calculus (differentiation and integration).