# Technical Task for Data Scientist

The dataset contains medical information about patients and their health. The task is to predict the likelihood of a stroke based on various factors.

**Dataset description:**

- id — patient identifier.
- gender — gender
- age — age of the patient
- hypertension — presence of hypertension
- heart_disease — presence of heart disease
- ever_married — whether the patient has ever been married
- work_type — type of work
- Residence_type — type of residence
- avg_glucose_level — average glucose level.
- bmi — body mass index
- smoking_status — smoking status
- stroke — target variable

**Task:**

- **Data Exploration:**
  - Conduct an exploratory data analysis (EDA) to understand the structure of the dataset and identify key relationships between variables.
- **Data Preprocessing:**
  - Handle any necessary preprocessing, if it's needed, to prepare the dataset for modeling.
- **Model Building:**
  - Split the data into training and test sets, and use any ML model to predict the likelihood of stroke (**you must try at least one bagging and one boosting model**)
  - Choose the best model.
  - Evaluate the performance of your model using appropriate metrics.
- **Feature Interpretation:**
  - Analyze the importance of features to interpret the model's predictions.
- **Conclusion:**
  - Summarize your findings, suggest improvements for the model, and provide any recommendations based on the analysis.
- **Prepare scripts and files:**
  - Read on next page

**Evaluation Criteria:**

- Ability to perform exploratory data analysis and identify key patterns.
- Data preprocessing skills, including handling missing values and categorical features.
- Understanding of gradient boosting methods and their application to classification tasks.
- Ability to interpret the model and explain feature importance.

# Notes

- All your work should be done using Python, all necessary libraries and Jupyter Notebook/Lab or Google
- As a result, you should have *.ipynb file with all your code and research

# Prepare script and files

- Save your model as a local file(and other files, if you need), using joblib library
- Prepare Python script that:
    - loads local model from file
    - loads features from file, filename should be set through terminal argument (take ‘test’ part of data, **but use unprocessed (original) version of data**)
    - makes predictions (result should be the same, as in your Notebook)
    - saves predictions with record ID and predicted value as a local *.csv file
    - the script should be launchable from the command line or terminal. Command:

      `> python <script> <input>`

  where:

  <script> - Python script file

  <input> - file with inputs (features)

  Note: model file should be loaded from inside script
- *you don’t need to train model again in script, you just need to load local modal from file*
- As a result, with *.ipynb file with research, you should also have model file, *.py script file, *csv file with inputs for model, *.csv file with model predictions