Task will contain two parts: **basic** and **CV** related.
In the **basic part** we assume that regression and classification tasks will be done with usage of simple neural networks. As soon as datasets used for these tasks have already been used in previous homeworks similar EDA and cleaning may be done.
In the **CV** related part we will implement binary classification which should classify images in 2 classes - hot dog or not. Dataset already separated on train and test part.
This article might help with a solution: link - ***this post was originally written in June 2016. It is now very outdated. If you want to use it as an example, you will have to make some fixes there.***

Overview
Basic part:

- Regression task - https://drive.google.com/drive/folders/13zKKNCa5srQSC8pqYWCIn5oRPXMlOGid?usp=sharing
  (same dataset with regression 1 module)
- Classification task - https://drive.google.com/drive/folders/1UF9FPkA06W7Z2rK3U2oDnkW3phq0VSYg?usp=sharing
  (same dataset with classification 1 module)

CV part:

- Dataset - https://drive.google.com/file/d/1zew1CtclL_AJSpPTpNTR84mQOOeiHbF3/view?usp=sharing
- Binary classification
- Images are splitted on 4 groups - train/test, hot dog/not hot dog
- Metrics: Accuracy
- Build a model for classifying images in two classes, use this example as a basis.

Evaluation
Criteria Scoring (36 max):

- **Basic part** - 10 points max;
  - Regression task solved - 2 points;
  - Classification task solved - 2 points;
  - Compare performance with the previous approaches (from *regression-1* and *classification-1*) - 2 points;
  - **Bonus:**
    - At least two different frameworks used for regression task:
      - Tensorflow/Keras - 1 point;

            - PyTorch - 1 point;
        - At least two different frameworks used for classification task:
            - Tensorflow/Keras - 1 point;
            - PyTorch - 1 point;
- **CV part** - 16 points max:
    - Baseline model with convolution layers trained - 1 point;
    - Different image augmentations used - 1 point;
    - Use pre-trained model - 3 points;
    - Un-freeze pre-trained model layers - 2 points;
    - Visualize learning curves(for Tensorflow/Keras **OR** PyTorch) - 2 points;
    - **Bonus**:
        - Compare performance of different pretrained models - 2 points;
        - Find optimal learning rate - 1 point;
        - At least two different frameworks are used for the task:
            - Tensorflow/Keras - 2 points;
            - PyTorch - 2 points.
- **General criteria for whole work** (10 points max):
    - The presence of informative visualizations (the presence of headings, axis labels, legends, etc.) - 2 points;
    - Explanations and comments regarding data actions performed - 2 points;
    - Active code reuse (helper methods, classes, etc.) - 3 points.
    - The presence of a clear structure in the work (introduction, separate sections on the issues under investigation, conclusion, further steps to take) - 3 points;