# Coding Task Description

Design and implement a pipeline for calculating category predictions with given ML-model. The pipeline is intended to be part of the other analysis flow and it should define clear input and output APIs. The design should allow easy scalability.

There is a lot of freedom to choose software technologies, tools and file formats to achieve the goal.

Note that the task is meant for evaluating both coding skills and software architecture design skills. Pay attention to the design of applications you create. Be ready to defend your architectural decisions in a discussion.

It is necessary to personally write the source code, but it does not have to be complete.

## Main functions:

- Your solution provides full pipeline: input -> model -> output.
- Input data can be coming from multiple sources, one sample at a time.
- There should be clear API for other components to use.
- Category classification is calculated with given model for each sample.
- It is easy to replace the used model with new one.
- Output label and processing time of each sample is recorded for logging and visualization purposes.
- It is possible to send original sample and the prediction for further processing to other components.

## Details:

- Author of the model wants to see the performance metrics of the model. Provide common measures and visualizations about performance and precission.
- You can consider the given Model to be a black box.
- Model is provided in Python Pickle format, it was build using Python 3.5.1 and Anaconda 2.4.1 (64-bit).

## Input data:

- For sake of simplicity the data is from publicly available Iris data set (https://archive.ics.uci.edu/ml/datasets/Iris).
- It's enough to use provided `example.json` data file for testing. The provided data file contains list of records of format:

```
{
  "id": 0,
  "label": 0,
  "info": [
    4.9,
    3.1,
    1.5,
    0.1
  ]
}
```

where field label is optional. If present, it tells the expected category for the entry.

**Model API:**

- See provided `Model.py`

## Deliverables:

The software is delivered with the full source code included. All used source code must be freely distributable.

If necessary, include readme.txt to describe the software and how it meets the requirements.

## Design question

Assuming that the number of samples increases, how does your system scale? What kind of monitoring/metrics you would include to the production system?