INFINUM

+385 91 606 4062
hello@infinum.com
infinum.com

ASSIGNMENT FOR MACHINE LEARNING ENGINEER POSITION

In a production process of bricks there are 4 sensors (s_1, s_2, s_3, s_4) which monitor the quality of each product. It is known that the sensor outputs are in a complex relationship with two important characteristics of brick:

- Compressive Strength (c)
- Flexure Strength (f)

Company ran some tests and measured those characteristics on a limited set of bricks. Data recorded from measurements and sensors is available in the file *dataset.csv*. Each row in the dataset contains 6 numbers s_1, s_2, s_3, s_4, c and f. We want to predict a value of c and f in order to detect bricks which do not satisfy minimal required values. Prediction needs to work in real time, so performance of the system is important.

Task 1

Train a neural network (Keras or Tensorflow) which is going to predict values of c and f based on outputs from sensors. Use just two hidden layers with 20 neurons and Mean Absolute Error (MAE) loss function.

Task 2

Create a Docker image which is going to serve the model. Use Tensorflow ModelServer for serving the trained model.

Task 3

Define AWS architecture and service required to deploy the model. Describe it with textual notes or a graph. Keep in mind data privacy and protection of intellectual property.

Task 4

Try to improve model accuracy and compare results with the initial version. You can assume that:

- $c = s_1 + s_2^2 + 2s_4$
- $f = s_1 + \frac{s_2}{2} + 3s_4 + 30$

Expected sensor values are:

$$s_1 \in [0, 10], \qquad s_2 \in [-5, -3], \qquad s_3 \in [0, 1], \qquad s_4 \in [20, 50]$$

For example, you can generate additional training data.
Create a new Docker image which is going to serve both models.

Important notes:

- Do not focus on the precision of the model. Focus on clean and simple implementation.
- In *Task 4,* focus on comparison of the results. New model doesn't need to have better performance.