MACHINE LEARNING SERVER

Backend Engineer Take Home Test

# Problem Overview

You are tasked to build a simple machine learning web server that is capable of training a model and predicting using it. One can think of it as a very simple DataRobot. No prior machine learning experience is required to complete the task.

# Task

1. Create a web server (preferably Flask) that is able to create a machine learning project, train a model and make predictions on new data.
2. Only two endpoints are required:
   * POST /create?target=<column\_name> – main endpoint, it has to do the following:
     1. Take a CSV dataset file using form param csv\_file
     2. Train a model on full dataset using column\_name column as a target
     3. Save model to disk so it can survive web server restart (replace the existing model if there is one already).
     4. Return HTTP 200 OK with an empty body
   * POST /predict **-** prediction endpoint, it has to do the following:
     1. Take a single CSV line from form field input\_line
     2. Load a model (return HTTP 404 Not Found if there is no model)
     3. Predict on a given line
     4. Return HTTP 200 OK with the predicted class in body (integer or categorical)
3. As a model, please use Support Vector Classifier from [scikit-learn](https://scikit-learn.org/stable/modules/svm.html). We do not grade model accuracy/performance in this test, it just has to make predictions later.
4. Server only works with a single model at a time, no need to implement users/authentication. Database is not required.
5. Support classification only. Only numeric columns.
6. **Functional tests using pytest should be present**
7. All code should be PEP8 compliant.

# Data Example

The data that you can use for testing is **iris** dataset. Download the dataset from the following URL:

<https://s3.amazonaws.com/datarobot_public_datasets/iris.csv>

The data dictionary is given below. Your classification target is **Species**:

| **Column Name** | **Type** |
| --- | --- |
| Sepal.Length | Numeric |
| Sepal.Width | Numeric |
| Petal.Length | Numeric |
| Petal.Width | Numeric |
| **Species** | **Categorical** |

# Usage

We will test your project in the following way (numbers may be different):

curl -X POST -F 'csv\_file=@iris.csv' 'http://localhost:5000/create?target=Species'

expecting 200 OK with an empty body

curl -X POST -F 'input\_line=5.1,3.5,1.4,0.2' 'http://localhost:5000/predict'

> setosa

expecting 200 OK with one target value in the body

# Submission

Implement your solution as a Python script using Python 3.6 or above. Make sure the results are reproducible.

Submit a **zip archive** with your source code and comments to the URL mentioned in the email.

Please submit the test within 3 days after receiving it.

# Questions and Clarifications

The test is designed to be completed in about one evening. Some of the tasks have a lot of room for experimentation and iterative improvement, but we do not expect state-of-the-art implementation, so a baseline implementation is OK. If you have any ideas for further improvement, you can mention them in a readme file.

Two of our engineers will be at your disposal ([anatolii.stehnii@datarobot.com](mailto:anatolii.stehnii@datarobot.com), [anastasiia.tamazlykar@datarobot.com](mailto:anastasiia.tamazlykar@datarobot.com)) while you are working on the task. Note that they can answer during business hours only.