

1. Counting islands. *Classical Algorithms*

You have a matrix $M \times N$ that represents a map. There are 2 possible states on the map: 1 - islands, 0 - the ocean. Your task is to calculate the number of islands in the most effective way. Please write code in Python 3 and **provide a github repository with a solution.**

Take care that you clearly understand test cases before starting implementation.

Inputs:

M N

Matrix

Test cases:

Input:

3 3

0 1 0

0 0 0

0 1 1

Output: 2

Input:

3 4

0 0 0 1

0 0 1 0

0 1 0 0

Output: 3

Input:

3 4

0 0 0 1

0 0 1 1

0 1 0 1

Output: 2

2. Regression on the tabular data. *General Machine Learning*

You have a dataset (train.csv) that contains 53 anonymized features and a target column. Your task is to build a model that predicts a target based on the proposed features. Please provide predictions for the hidden_test.csv file. Target metric is **RMSE**.

The main goal is to provide github repository that contains:

- jupyter notebook with exploratory data analysis;
- train.py python script for model training;
- predict.py python script for model inference on test data;

- file with prediction results;
- readme file that contains instructions about project setup and general guidance around project;
- requirements.txt file.

Please provide documented code. Scripts (train.py and predict.py) should be able to be executed from the terminal.

3. MNIST classifier. *OOP*

You have 3 different models to solve MNIST (handwritten digits database) classification problem:

- Convolutional Neural Network (any architecture, any framework);
 - ***Input:*** tensor 28x28x1;
- Random Forest classifier;
 - ***Input:*** 1-d numpy array of length 784 (28x28 pixels);
- Model that provides random value (for simplicity) as a result of classification;
 - ***Input:*** 10x10 numpy array, the center crop of the image.

The goal is to build a ***DigitClassifier*** model that takes an ***algorithm*** as an input parameter. Possible values for the algorithm are: ***cnn***, ***rf***, ***rand*** for the three models described above.

There is **NO** need to implement a training function inside ***DigitClassifier*** and focus on the quality of the model, just raise a *Not implemented exception*. We need to focus only on the predict function that takes a 28x28x1 image as input and provides a single integer value as output.

Ideally, the solution should contain:

- Interface for models like Convolutional Neural Network, Random Forest classifier, Random model. Potentially other developers will develop new models, so we need to have an interface for them. Let's call it ***DigitClassificationInterface***.
- 3 classes (1 for each model) that implement ***DigitClassificationInterface***.
- ***DigitClassifier***, which takes as an input parameter the name of the algorithm and provides predictions with exactly the same structure (inputs and outputs) not depending on the selected algorithm.

Please provide a github repository with your solution.