Assignment 3

ML in Statistics, Argyn Kuketayev ©2022-2024

A customer wants to map the terrain of a square plot of land. You are given the candidate locations of the probes and asked to predict the altitudes of the terrain at probe locations. To predict the altitudes, you must train an ensemble regressor of your choice using the terrain dataset that is provided to you.

# Introduction

The plot of land is square 100x100, and its terrain features a landmass with altitude greater than 0, and lakes with altitude zero. For instance, the terrain may look like in the following plot, where the dark blue areas with altitude zero are lakes.

Chart

Description automatically generated

You are given a dataset with irregular coordinates X and Y and their altitudes Z, e.g. as marked on the following chart with dots.

Diagram

Description automatically generated

*Figure 1. An interpolated plot of a terrain sampled by irregular coordinates x,y and corresponding altitude z.*

You are to build the regression model then train it on this data so that it can predict for any given point on the land with coordinates x,y its altitude z. You will then use this regression model to draw the plot of the land.

The plot of lakes may look as in the following figure.

A picture containing icon

Description automatically generated

*Figure 2. Plot of the lakes of the example terrain*

There are four graded tasks in the assignment plus one optional ungraded task to complete.

# References

* GDrive directory seed0 with example data and plot:   
  [seed0](https://drive.google.com/drive/folders/1r-wFZaSkJq6UrWjBcZv3MelHQbE7D6Rg?usp=sharing)
* A terrain dataset [terrain\_3.csv](https://drive.google.com/file/d/1GJ1zQb_EK1CV_4ZT2mIHXeRcM_qFhtBt/view?usp=share_link)
* A file with candidate probe locations [lake\_probe\_candidates\_3.csv](https://drive.google.com/file/d/15kgozPNj2EwIkrgkE3cwJxCzYSll80tg/view?usp=share_link)
* Google Colab Notebook with an assignment Python code template: [mlwp Ass 3 template.ipynb](https://colab.research.google.com/drive/1topOIYE7unRA_iLjJBZO3MZnL-q2hWY0?usp=sharing)

# Setup

In your Google Drive create the working directory for the assignment   
‘My Drive/Colab Notebooks/mlclass/ass3’

Click on the url of seed0 directory in the References section, then in the dropdown pick “Add shortcut to Drive” and select the working directory to save the shortcut.

Create a directory called seed20 in the assignment GDrive. Inside this directory, create the shortcuts to terrain\_3.csv and lake\_probe\_candidates\_3.csv files.

Click on the url of the assignment template notebook in the References section. Save the notebook in the working directory. **Do not** create the shortcut to Drive in this case.

Your working directory should be similar to this figure (replace 2 with 20):

Text

Description automatically generated with medium confidence

In the remainder of the assignment, you must modify the Python notebook template to complete it.

Run all steps of the Assignment before you reach the Task 0. One of the code sections, “Load Terrain data”, loads the terrain data. The terrain is sampled at irregular points, and the file may look as follows:

Table

Description automatically generated

Columns X and Y are coordinates of points, and Z is the altitude. Whenever Z==0 the point is on the lake.

The probe candidate points are in a similar tabular file, but they don’t have a column Z, as shown on the next figure. Naturally, your customer doesn’t know the altitude of these points and is asking you to predict them.

Table

Description automatically generated

Pay close attention to the “Example terrain…” section of the notebook as it may give you ideas on how to complete the tasks of this assignment. In this section you are shown how to plot a known terrain given in a regular grid, something that you have to build yourself in the tasks for your terrain.

# Global Variables

The template uses several variables across multiple cells:

* xa, xb, ya, yb – boundaries along x and y coordinate axes of the square plot of land
* Input file names (paths):
  + terrain\_file
  + probe\_file
* Input example file names (paths):
  + ex\_plotdata\_file
* Output file names (paths):
  + probe\_out\_file
* Pandas dataframes corresponding to input terrain dataset and the probes locations:
  + df\_terrain
  + df\_probes
* x, y, z – coordinates and altitudes, that are extracted from terrain\_file

# Task 0 (0 points), Optional.

Although the task is optional, it is a useful exercise not only for your learning to plot surfaces of irregular terrain, but also to get an idea of what kind of terrain you are working with before even starting to build the terrain model. For instance, in the example terrain given to you in seed0 directory, you can plot the heatmap of the terrain by using the interpolation as shown in Figure 1. You may compare the terrain with what your classifier is predicting to get an idea of whether the training is going on as expected.

# Task 1 (6 points)

This is the main task of the assignment. You must train an ensemble regression method on the data given in terrain\_3.csv file. You must code the function fit\_model, which returns the trained (fitted) regressor class reg and the vector zhat with predicted altitudes for the x,y coordinates given in the terrain data (in sample predictions).

You can use any method that we covered in the ensemble methods lecture. You can manually pick the hyperparameters of the model or use one of the hyperparameter optimization techniques that we discussed in this course, e.g. grid search cross validation. You may receive 6 points for this task only if hyperparameter optimization was used, otherwise you can get only up to 5 points.

You may want to build a pipeline, and use the feature scaling or other transformations.

# Task 2 (1 points)

You must calculate the root mean squared error (RMSE) for the terrain data by comparing predicted altitudes in zhat (from Task 1) with the true altitudes in the terrain dataset.

# Task 3 (2 points)

You must plot the terrain similar to the plots below in this section. Use a function plotland that is given in the “Useful Functions” section of the Python template. Examples of its use are given in the section “Example terrain and its plots” of the notebook. The idea is to first build a meshgrid X,Y where these are two dimensional arrays corresponding to coordinates on the square land. Read the instructions in the task template. An example meshgrid is given in X0,Y0 arrays of the example terrain. You can use the numpy package’s meshgrid function.

Once you have the X,Y grid, you will run your regression model to create Z arrays that correspond to these grid coordinates. Z will be the predicted altitude at the X,Y coordinates.

Without hyperparameter optimization, your terrain plot may look like the figure below for the example data set:

Map

Description automatically generated with low confidence

*Figure 3. Predicted contour plot of the example terrain, no hyperparameter optimization applied*

With hyperparameter optimization you may be able to plot a more accurate contour plot as the next figure for the example terrain:

A picture containing text, monitor, screen, display

Description automatically generated

*Figure 4. Predicted contour plot of example terrain*

# Task 4 (1 point)

You must predict the altitudes for the coordinates given in lake\_probe\_candidates\_3.csv file. Your predictions are saved in a file probe\_predictions.csv.

The file may look like the next figure.

Graphical user interface, application, table, Excel

Description automatically generated

# Submitting the assignment

Before submitting the assignment make sure to select ‘Runtime>Restart and run all’ in Colab’s menu. This is to ensure that your code runs from scratch in a new session. Print the solution notebook to PDF, checking that all code, text outputs and images are present in the PDF. Download your notebook from the File menu in Colab as .ipynp file. Submit **both** the notebook and its PDF to BlackBoard.

Finally, submit a file probe\_predictions.csv with your prediction for candidate probe locations.