Welcome to Task 2 (Gradient Prediction)’s documentation!

Contents:

task2 main

This script is (please insert the question here) ...

*class*task2.**Data**(*name*, *m=5*, *std=0.4*, *num\_samples=100*, *x\_factor=1*, *files=False*,

*plots=False*)

This is the test data generation class

It is used to generate arbitrary sample data with specified properties given by the user.

Parameter:

* name (str): the name associated with the generated data
* m (float): the gradient of the generated data. Default value is 5
* std (float): the standard deviation of the generated data. Default value is 0.4
* num\_samples (int): the sample size of the generated data. Default value is 100
* x\_factor (float): the x-scaling factor of the generated x-data. Default value is 5
* files (bool): variable specifies whether the data files should be saved. it saves if its value is true. Default value is False
* plots (bool): variable specifies whether the plots should be saved. it saves if its value is true. Default value is False

**evaluate**()

This method predicts gradient using curve fitting

It computes the gradient and y intercept of given data It also generated the predicted line

Parameter:

None

Returns:

None

**line**(*x*, *M*, *C*)

This method computes line data from line equation parameters for fitting

Parameter:

x (np.array): x-data samples M (float): gradient of a straight line C (float): y-intercept of a straight line

Returns:

y (np.array): y-data samples

**preprocess\_list\_line**(*data*)

This method is used to format y-data

Parameter:

data (list array): x-data and y-data pair

Returns:

return a list of y-data from generated data pair

**save\_plot**()

This method is used to save plots as images in the image directory

Parameter:

None

Returns:

None

**write\_xls**()

This method is used to save data as an excel files file

Parameter:

None

Returns:

None

*class*task2.**Test**(*parent=None*)

This is a class for running generated data tests

it runs an n x m factorial experiment on various slope and gaussian noise ranges and renders the results as a 2D or 3D plot. You also have the option of saving the plots to an image directory of the data to a file directory in a .xslx format.

Parameter:

parent (object): The parent class of the test class

**createButtons**()

This method is used to create the graphic user interface forms.

Parameter:

None

Returns:

None

**error**(*msg*)

This is the method that handles error messages

Parameter:

msg (str): The error message to be displayed

Returns:

None

**info**(*msg*)

This is the method that handles informative messages

Parameter:

msg (str): The information to be displayed

Returns:

None

**plot**()

This method handles all plots.

It plots in 2D if that option is checked otherwise the plot is rendered in 3D.

Parameter:

None

Returns:

None

**plot2D**()

This method handles 2D plots

Parameter:

None

Returns:

None

**plot3D**()

This method handles 3D plots

Parameter:

None

Returns:

None

**run\_test**()

This method runs the curve fitting test on data.

it runs an n x m factorial experiment on various slope and gaussian noise ranges and renders the results as a 2D or 3D plot.

Parameter:

None

Returns:

None

*class*task2.**Window**(*parent=None*)

Main window form class

Parameter:

parent (object): The parent class of the window class

**createButtons**()

This method is used to create the graphic user interface forms.

Parameter:

None

Returns:

None

**error**(*msg*)

This is the method that handles error messages

Parameter:

msg (str): The error message to be displayed

Returns:

None

**info**(*msg*)

This is the method that handles informative messages

Parameter:

msg (str): The information to be displayed

Returns:

None

**line**(*x*, *M*, *C*)

line equation for fitting

**plot**()

This method handles all plots.

Parameter:

None

Returns:

None

**preprocess**(*data*)

return a list of Y data from read file

**read\_file**()

This method reads a .txt file form source folder

Parameter:

None

Returns:

str: the data in the read file

**read\_xls**(*name*)

This method reads an excel file from a specified path

Parameter:

name (str): specifies the path and name of the excel file to be read

Returns:

None

**runtest**()

This method computes line data from line equation parameters for fitting

Parameter:

None

Returns:

None

**select\_file**()

This method helps the user to a select source data file

Parameter:

None

Returns:

None

**write\_xls**(*data*, *name*)

write data to excel file path

Parameter:

data (list): data to be written name (str): specifies the path and name of the excel file to be written

Returns:

None



Fig 1. Flow chart of Test Running Operation



Fig 2. Flow Chart of Gradient from file prediction operation