Please submit a single document for this assignment on Canvas by the beginning of class on **Monday, Oct. 28th**.

This is a short assignment in which we check whether our trained non-linear model from Assignment 6 *generalizes* well. In other words, we want to gauge how well our trained model performs on examples that it did not see during training.

Important:

- In Assignment 6 you should have found a model architecture (number and widths of layers along with nonlinearities) and associated learning parameters (batchsize, learning rate, momentum, and epochs) that lead to an explained variation proportion significantly higher than 0.86. **Redo Assignment 6 if that is not the case.**

- **Use that same model architecture, those same nonlinearities, and those same learning parameters for this assignment.**

We now want to test how well your trained model generalizes:

- (Randomly) partition our original dataset into *train* and *test* sets. Use variables names similar to `xss_train` and `xss_test`; of course, you will also define the corresponding outputs `yss_train` and `yss_test`.

  In practice, one often uses an 80-20% split. On our current housing dataset, you might consider training on 2000 examples and testing on 264.

  Now **train your model using just the training data.**

- The last code block in your program for Assignment 6 is likely the following:

```
# Compute 1-SSE/SST which is the proportion of the variance in the data
# explained by the regression hyperplane.
SS_E = 0.0;   SS_T = 0.0
mean = data.mean(0)[0] # mean of the outputs (zero, if data are mean-centered)
for datum in data:
  SS_E = SS_E + (datum[0] - model(datum[1:]))**2
  SS_T = SS_T + (datum[0] - mean)**2
print('1-SSE/SST = {:1.4f}'.format(1.0-(SS_E/SS_T).item()))
```

  Replace that entire code block with something similar to:

```
print("explained variation:", dulib.explained_var(model, (xss_train, yss_train)))
```

  Add a line that prints out the proportion of variation explained by your model on test data.
  Submit:

  - a screen shot of a run of your program.
  - a screen shot of the part of your program where you split the dataset into train and test sets.

  The above assumes that you have installed DUlib. If you haven't already, issue this command at your command line:

```
pip install dulib --user
```

  Then import DUlib into your program like this: `import du.lib as dulib`