Example 1: ML with continuous data

A. References

This guide is based on a the study Design of experiments for optimizing the calendering process in Li-ion battery manufacturing, with the raw data found here.

B. Data.xlsx

For this example, the sample Data.xlsx can be found at `\docs\examples and guides\Example1\Data.xlsx`. Copy this file into `src`.

`Train Data` tab

This tab contains the raw data. Each row is an experiment or run and each column represents a type of data.

`Test Data` tab

Same as the `Train Data` tab but contains the data used for validation.

`Design Parameters` tab

This contains the table used to identify which columns in `Train Data` and `Test Data` are used as features (i.e. inputs to the model). Each row is a separate feature. The table contains the following columns:

- **Code**: code or ID for each feature. Best practice is to have these as single letters (i.e. A, B, C, etc.)
- **Features**: *exact* column name in `Train Data` or `Test Data`.
- **Feature type**: either `Numerical` or `Categorical`. Determines whether the data type is a continuous numerical value or will be treated as categorical data.
- **Min Level**: minimum value for the scaled model to be mapped to -1 after scaling.
- **Max Level**: maximum value for the scaled model to be mapped to +1 after scaling.
- **Term type**: either `Process` or `Mixture`. Code currently only accepts `Process`.

`Responses` Tab

This contains the table used to identify which columns in `Train Data` and `Test Data` are used as responses (i.e. outputs of the model). Each row is a separate response. The table contains the following columns:

- **Response*: *exact* column name in `Train Data` or `Test Data`.
- **Lambda**: lambda values (comma-separated) to use in power transformations. Recommended to use the default values of `-2,-1,-0.5,0,0.5,1,2` unless there is a reason to select more specific values.

`Misc` Tab

Contains misc. info. The user won't need to interact with this tab.

C. auto_mlr.py

After filling out Data.xlsx, open `src\auto_mlr.py`.

Enter the list of terms to be used in the model in `terms_list` under the `USER DEFINED INPUTS` section. Use the info entered under the `Code` column in the `Design Parameters` tab in `Data.xlsx`. Each term follows the patsy format. In this example, we are using the model shown in Eq. 1 in the paper:

$$y^{\lambda} = \beta_0 + \beta_A*A + \beta_B*B + \beta_C*C + \beta_{AB}*A*B + \beta_{AC}*A*C + \beta_{BC}*B*C + \beta_{A2}*A^2 + \beta_{B2}*B^2$$

meaning we set

```
terms_list = ['A', 'B', 'C', 'A:B', 'A:C', 'B:C', 'I(A**2)', 'I(B**2)']
```

In this example, `A`, `B`, and `C` represent the linear terms, `A:B`, `A:C`, and `B:C` represent the 2-feature interaction terms, while `I(A**2)` and `I(B**2)` represent the non-linear quadratic terms. Note that there is no $\beta_{C2}*C^2$ term in Eq. 1, meaning it was also not added to `terms_list`.

Afterwards, run `auto_mlr.py`.

D. Output folder

After running `auto_mlr.py`, 3 items in the `src\Output` folder:

- **Box-Cox (folder)**: folder containing Box-Cox plots. Each figure represents the Box-Cox plot for a single response. Lambda values below the dotted blue line represent the lambda values which are within the 95% confidence interval.
- **Pred vs Act (folder)**: folder containing predicted vs actual plots. The filenames are written as `{lambda}_response.jpg`. Black points are from the training set while red points are from the testing/validation set.
- **Models summary.xlsx**: Excel file containing info on all the models. More details in the next section.

Models summary.xlsx

Each tab contains different information on the models, specifically:

- **all models**: contains all the information about each model fit. Model terms are in *encoded* units.
- **best models**: the best models from `all models` are selected and shown here. The logic is described in the `get_better_model` function in `src\mult_lin_reg_utils\model_reduction.py`.
- **all models in real units**: same as `all models` but terms are in real units.
- **best models in real units**: same as `best models` but terms are in real units.

Making predictions using the models

To use a model, take any model from `Models summary.xlsx` and create the model formula from the terms in the sheet. For example, in the `best models in real units` tab, we can take the terms of the `_10CmAhg` model and plug them into Eq. 1 above to get

$$y^{1} = -459.49 + 40.22*B - 1.03*C - 0.60*C^2$$

We can then input values of B (porosity in %) and C (density in g/m^2) to the equation above to calculate the capacity at 10 C (in mAh/g).