Statistical Machine Learning – Week 4

Sascha Jecklin

HS 2022/23

1 Task

In this lab we will implement train several linear regression models using `scikit-learn` and `statsmodels` functions / methods. Use the template `week4_exercise.py` and complete all ToDos. The data loading is already handled in the `main()` function by using `pandas read_csv` method. `Pandas` is often used in Machine Learning to load, preprocess and display data and it stores the data in `Series` or `DataFrames`. Libraries like `scikit-learn` and `statsmodels` also accept pandas DataFrames. For more information on pandas and its methods, see this link.
Get an overview of the script `week4_exercise.py` and finish the following TODO's.

- Finish the function `lin_reg_with_sklearn(x, y)` by using sklearn's LinearRegression class. The function is called with the Advertising data set and it should return the intercept and slope of the simple linear regression of sales onto TV budget.
- Complete the function `reg_with_smf(data, formula_str: str)` by using statsmodels methods. Again, this function should return the intercept and slope of a simple linear regression of sales onto TV budget. In this function also print the summary table which contains the statistics on the model, similar to Table **??**. Hint: see this link.
- Implement the `my2dplot(x, y, intercept, slope)` which takes the data set and the model parameters as arguments and displays a scatter-plot of the data as well as the regression line.
- Use the `reg_with_smf` function to make a multiple linear regression of sales onto TV budget, radio budget and the interaction between TV and radio budget. Analyse the summary table and comment on the significance of the predictor variables.
- Print a table which contains the correlation coefficients between all predictor variables in the advertising data set. Hint: use an in-built pandas function.

If there is something unclear, have a look at the documentations of sklearns LinearRegression and StatsModels OLS. Verify your results using the following tables

	Coefficient	Std. error	t-statistic	p-value
`Intercept`	7.0325	0.4578	15.36	< 0.0001
`TV`	0.0475	0.0027	17.67	< 0.0001

Table 3.1: Summary table for regression of sales onto TV. Copied from page 68 in the book.

	Coefficient	Std. error	t-statistic	p-value
Intercept	6.7502	0.248	27.23	< 0.0001
TV	0.0191	0.002	12.70	< 0.0001
radio	0.0289	0.009	3.24	0.0014
TV×radio	0.0011	0.000	20.73	< 0.0001

Table 3.9: Summary table for regression of sales onto TV, radio and interaction. Copied from page 88 in the book.

	TV	radio	newspaper	sales
TV	1.0000	0.0548	0.0567	0.7822
radio		1.0000	0.3541	0.5762
newspaper			1.0000	0.2283
sales				1.0000

Table 3.5: Correlation coefficient between all predictor and response variables. Copied from page 75 in the book.

Comments

- Matplotlib works similiar to `plot` in MATLAB. Check `https://matplotlib.org/3.1.1/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py` for some examples.
- To convert a pandas DataFrame into a numpy array, use `DataFrame.to_numpy()`. If you want to access the observation values of a particular feature in a pandas DataFrame you can do this with `advertising.`*`Feature`*. For example `advertising.TV` or `advertising.sales`.
- StatsModels OLS modules accepts inputs in the form of *Patsy formula language description*. This is often used in *R*. The method takes the formula as a string. For a model with three predictors `x1`, `x2`, and `x3` the string has the form `'y ~ x1+x2+x3'`.
- You do not have to calculate standard errors, t- and p-values. StatsModels offers a method for this.
- numpy arrays can be reshaped with the method `reshape()`. reshape() takes the new dimension as input. All elements of the array need to fit into the new shape. numpy allows to give one of the shape parameters as -1. It means that it is an unknown dimension which numpy has to figure out. For example from shape (3,3) to (9,1) or (-1, 1).
- If a module is not already installed use the command `pip3 install` *`module`* `--user` to install it.