Intermediate Machine Learning with scikit-learn

Pandas Interoperability, Categorical Data, Parameter Tuning and Model Evaluation

By Thomas J. Fan

Link to slides

Scikit-learn is a Python machine-learning library used by data science practitioners from many disciplines. We will learn about Pandas interoperability, categorical data, parameter tuning, and model evaluation. For Pandas interoperability, the ColumnTransformer applies data transformations on different columns from a Pandas DataFrame. In version 1.2, all of scikit-learn's transformers are configurable to output Pandas DataFrames. Next, we will learn about categorical data and how to use scikit-learn's encoders to convert these categorical features into numerical features for a machine learning algorithm to consume. We will explore tuning algorithms in scikit-learn with grid search and random search. Model evaluation is an essential part of the machine learning workflow. We will cover the metrics provided by scikit-learn and how to use the scoring API. Furthermore, we will use the plotting API to visualize a model's performance. Finally, we use all the ML techniques we learned to train and evaluate a model on a house pricing dataset with Histogram-based Gradient Boosted Trees.

Obtaining the Material

With git

The most convenient way to download the material is with git:

```
git clone https://github.com/thomasjpfan/ml-workshop-intermediate-v2
```

Please note that I may add and improve the material until shortly before the session. You can update your copy by running:

```
cd ml-workshop-intermediate-v2
git pull origin main
```

Download zip

If you are not familiar with git, you can download this repository as a zip file at: github.com/thomasjpfan/ml-workshop-intermediate-v2/archive/main.zip. Please note that I may add and improve the material until shortly before the session. To update your copy please re-download the material a day before the session.

Running the notebooks

Local Installation

Local installation requires conda to be installed on your machine. The simplest way to install conda is to install `miniconda` by using an installer for your operating system provided at docs.conda.io/en/latest/miniconda.html. After conda is installed, navigate to this repository on your local machine:

```
cd ml-workshop-intermediate-v2
```

Then download and install the dependencies:

```
conda env create -f environment.yml
```

This will create a virtual environment named `ml-workshop-intermediate-v2`. To activate this environment:

```
conda activate ml-workshop-v2
```

Finally, to start `jupyterlab` run:

```
jupyter lab
```

This should open a browser window with the `jupterlab` interface.

Run with Google's Colab

If you have any issues with installing conda or running `jupyter` on your local computer, then you can run the notebooks on Google's Colab:

1. Pandas output
2. Categorical Data & Pandas Input
3. Parameter Tuning
4. Model Evaluation

License

This repo is under the MIT License.