Machine Learning Project 2

Energy Prices Prediction

Goal of the Project

The aim of this project is to perform a regression task on time series (temporal data), and to become more familiar with the use of Multi-Layer Perceptrons. To do so, you will train an MLP classifier to predict energy prices in Germany using data on energy production per type (wind, solar, ...), energy demand, and many other features. This document introduces the data, as well as the different tasks that you are expected to perform.

Modalities

After performing the different tasks, you are expected to handle out a report of max. 4 pages. **Additional pages will not be read.** When writing your report, keep in mind that we want you to **justify your answers**, a properly written Python code is not enough (in other words, it is not necessary to give additional information about, e.g., how a model works, what is the dataset, the context, etc. but do provide justifications for your choices when relevant). This PDF report can be written in English or in French, and **should be exported in .pdf format.**

On the implementation side, you will use Python 3 with the `scikit-learn` framework (`http://scikit-learn.org`), which provides many tools for machine learning and has a detailed documentation. Please check the instructions in `https://scikit-learn.org/stable/install.html` to install `scikit-learn`. **Your code should be submitted with your report on Webcampus by providing a notebook or a script** (.ipynd or .py file(s)).

The deadline to hand in your PDF report and notebook/script on Webcampus is set to **Wednesday 26th November at midnight**.

!!! Please, carefully respect the aforementioned instructions. Do not hand in your documents in a .zip, and do not use any other formats than those specified !!!

The Dataset

In this project, you will train a basic regression model for baseline, as well as a MLP Regressor to predict futur prices of energy. To do so, you will use a historical dataset ranging hourly from January 1, 2018, to December 31, 2024. The raw dataset is made of 61,368 instances, each instance thus being a time series of 1 hour. The dataset contains a total of 27 features, and the one feature you want your model to predict is the **Historical Electricity Prices** feature. For more detailed information about this dataset, including the description and type of each feature, please refer to the following Kaggle: https://www.kaggle.com/datasets/datasetengineer/electricity-market-dataset/data

The dataset is **not** splitted into a training and testing set yet, which is the reason why you have only one .csv file. You can load it in a proper form using the pandas library.

Task 0 - Prepare your data

As a first preliminary task, you have to prepare your data according to the task you aim to perform. Once your .csv loaded, perform an exploration to investigate your data, get familiar with it. At this point in the project, you should ask yourself:

1. **Are all features relevant to use?** Some may have a negative impact on your model's ability to generalize if kept (Hint: think about data leakage, e.g. future data you don't normally have at the time you're trying to do the prediction = data "from the future"). Mention which ones you decided to drop in the report and explain why (Why would those have a negative impact on your model's generalization?).

2. **Is the "timestamp" feature usable as such in his current state?** (Hint: use wisely `pandas.to_date_time()`).

3. **Are all features kept compatible with a regression task?** (Hint: One Hot Encoding if you have categorical features).

4. **What is a relevant split for my training and testing set?** (Hint: a regular `train_test_split()` is not gonna cut it here...) Explain why and justify the chosen method in your report.

Task 1 - A Baseline Attempt

Before playing with MLPs, your first task will be to train and evaluate the performance of a simple Regression Model. Compute the training and testing

performances using the `score()` method. Add those performances to your report and briefly explain for your findings.

Train a linear regression model. What is the metric used by `scikit-learn` to evaluate the performance of the model? What can you conclude?

Task 2 - MLP Regressor

Your second task will be to train and evaluate the performance of a MLP regressor[1]. But before, quick reminder to **scale your data** (Hint: StandardScaler()) Make relevant choices for (at least) those meta-parameters and justify your choices in your report:

- `hidden_layer_sizes`
- `activation`
- `learning_rate_init`
- `max_iter`
- `alpha`

What would happen if, using the same meta-parameters, you set the activation function to `identity`? Briefly explain in your report.

Train your MLP. Justify your choice of meta-parameters. Report the training and testing performances using the `score()` method for both. Comparing your model with the `identity` one, which performs best? Explain the theoretical difference between the two activation functions.

Task 3 - Conclude your Findings

The goal of this last task, is to give us your (now) expert opinion on which model we should be working with to predict energy prices in Germany? Compare the linear regression, your MLP and the MLP with identity as activation function. Share your thoughts and conclusions, as well as whether or not your result confirms your first intuition or is counter-intuitive.

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html`