

I start ingesting the data provided in *train.csv* and parsing dates to datetime format using panda's `read_csv`.

Then I analyze the data, checking for null values and dealing with them. After realizing that there are entire rows with missing values, I drop those rows, making sure about not losing any data or breaking the data structure.

Then I select the independent variable features and the target feature (*Target*).

After that I proceed to split the data into training and test sets using sklearn's `TimeSeriesSplit` to not break the time series structure.

Later I define the F1 score metric to use (keras module doesn't have it by default).

Then I proceed to generate a Deep Neural Network using Keras and Tensorflow as a backend.

After manually checking some parameters (such as number of estimators, layers, dropout percentage, and optimizers) plotting scores and losses from the train and validation sets, I built a 3 layer network with 400 neurons, Rectified Linear Unit as activation and output layer with sigmoid activation, using batch normalization between layers.

Then I compile the model with the best parameters I found, using Early Stopping as a callback to minimize score losses and use it to predict the targets with the given *test_x.csv* file and save the results complying with the challenge's requirements.