Hackagig Challenge 1 UnicoRNN team submission notes

Notebooks are numbered according to stages of work

Data wrangling

1_0_data_wrangling.ipynb

Background and objectives to problem

Data exploration and data preprocessing focussed on the half hourly dataset. Pivots, merges etc mostly in Pandas.

Files were too big to work with in Feather format, .csv proved to be the most reliable (and slowest) format for >20GB files

Half hourly data was reconstructed from individual data files.

Some spurious non half hourly null value sample points were removed from the datset.

1_1_data_wrangling_db.ipynb

Due to memory (128GB RAM, 440GB swap) size limitation, half hourly data was imported into a PostgreSQL database for furhter SQL queries and export.

1_3_data_wrangling_4_6_a_forecast_NN_preprocessing.ipynb

A significant amount of data pre-processing for the 4_6_a_forecast_NN_daily.ipynb Pytorch Mixed Input Model using daily data was done.

The notebook was split into two parts, this being the first - preprocessing step.

1_4_data_wrangling_4_6_b_forecast_NN_hh_preprocessing.ipynb

Again, a significant amount of data pre-processing for the 4_6_b_forecast_NN_hh.ipynb Pytorch Mixed Input Model using half hourly data was done.

The notebook was split into two parts, this being the first - preprocessing step.

1_5_data_wrangling_consistens_ds.ipynb

The final data export from 1_4_data_wrangling_4_6_b_forecast_NN_hh_preprocessing.ipynb is used here for futher processing and export of a dataset for forecasting on individual households (see scripts folder)

Exploration

2_0_data_exploration.ipynb

Exploration of cross corellation, individual time series

Day vs count of households with daily 48 hours sampled data is very informative. Only a fraction of households have data spanning the entire date rangle. DHighest number of households with data spans ~ Nov 2012 through Feb 2014

2_1_energy_plots.ipynb

Exploration of daily repeat loads for individual households. Unlike commercial loads, the residential loads appear highly variable across past weeks on a pre day comparison basis

2_2_exploration_plots_ds.ipynb

Exploration of Acorn data - more could be done here to identify patterns in this dataset and correlate back to smartmeter data.

Correlation of income with power use (Acorns A and D are high income and biggest energy users, but Acorn B is also I high income group, energy useage not consistent with demand drom Acorn A and D - more investigation as to why is required.

2_3_household_data_exloration_ds.ipynb

Exploration of an individual household as well as discovery of a gap in the dataset.

A gap of 4 sample points in the half hourly dataset was discovered.

'2013-09-09 23:00:00', '2013-09-09 23:30:00', '2013-09-10 00:00:00', '2013-09-10 00:30:00'

Nearest interpolation was used to fill in this 2hr gap in the dataset.

Clustering

3_0_a_clustering_ds.ipynb

Several clustering methods tried.

Heirachical clustering - not that useful here are too hard to visualise >5500 households in this type of plot. Alternative representations (other than tree) of the clustering may be worth investigating.

Dynamic time warping paths - useful for looging at correlation betwee two time series

tsam code base - very rich representation of the time series energy use dataset.

Plots for 1132 households were made as input for imge-based-similarity clustering. See scripts/image-similarity-clustering/ for codes.

3_0_b_clustering_tsne_ds.ipynb

Results from image similarity clustering were plotted, Affluent acorns clearly clustering towards the left using the t-sne clustering algorith. Would be worth trying k-nn and other clustering algorithms.

3_1_clustering_kNN_ds.ipynb

Here K-nearest neighbours using Dynamic time-warping distance is used for clustering energy time series for different households.

RAM intensive and needs to be revisited on a smaller dataset.

Forecasting

4_0_forecast_daily_baseine.py

python script in scripts folder.

Baseline forecast using only historical daily values - one week ago, 1 year 1 week ago. If we cant beat this we need to re-visit our models.

This was run on only two households sampled from the same dataset as used for the 4_6_b_forecast_NN_hh.ipynb model

4_1_forecast_ARIMA_ds.ipynb

Stationarity, ARIMA algorithm forecast for 7 days using half hour dataset for single household. This code needs a re-vist to QC as ARIMA result seems inversely correlated to data.

4_2_a_forecast_ML_direct_day.py

python script in scripts folder.

multiple sklearn ML models

Walk-forward model validation.

Model is required to make a one week prediction, then the actual data for that week is made available to the model so that it can be used as the basis for making a prediction on the subsequent week.

This is both realistic for how the model may be used in practice and beneficial to the models, allowing them to make use of the best available data.

Code based on link below

See https://machinelearningmastery.com/multi-step-time-series-forecasting-with-machine-learning-models-for-household-electricity-consumption/

4_2_b_forecast_ML_recursive.py

python script in scripts folder.

Makes a multi-step time series forecast used recursively.

Make a prediction for one time step, feeding prediction the model as an input in order to predict the subsequent time step.

Repeat until the 7*48 step been forecast.

multiple sklearn ML models

Code based on link below

See https://machinelearningmastery.com/multi-step-time-series-forecasting-with-machine-learning-models-for-household-electricity-consumption/

4_4_a_forecast_GB_daily_ds.ipynb

LightGBN 7 day forecast model using daily dataset

Surprisingly easy to build and run this model, more time spent on feature engineering and ensembling may have impreved the result.

4_4_b_forecast_GB_hh_ds.ipynb

LightGBM 7 day forecast model using half hourly dataset.

Forecast prediction is for the last week of a 105 week 544 household subsampled dataset - exact same dataset as used for the 4_6_b_forecast_NN_hh.ipynb model

4_5_a_forecast_multichannel_cnn.py

python script in scripts folder.

Provide each one-dimensional time series of input variables to the model as a separate channel of input.

CNN uses a separate kernel and read each input sequence onto a separate set of filter maps, essentially learning features from each input time series variable.

Note we only use 3 input variables, cvould be extended to use more input data

Forecast prediction is for the last week of a 105 week 544 household subsampled dataset - exact same dataset as used for the 4_6_b_forecast_NN_hh.ipynb model

Code based on link below

see https://machinelearningmastery.com/how-to-develop-convolutional-neural-networks-for-multi-step-time-series-forecasting/

4_5_b_forecast_multiheaded_cnn.py

python script in scripts folder.

Multi-headed CNN model-Create separate sub-CNN model or head for each input variable

Defined a separate CNN model for each of the input variables.

Loop over each variable and create a sub-model that takes a one-dimensional sequence of n days of data and outputs a flat vector containing a summary of the learned features from the sequence. Each of these vectors can be merged via concatenation to make one very long vector that is then interpreted by some fully connected layers before a prediction is made.

Note we only use 3 input variables, cvould be extended to use more input data

Forecast prediction is for the last week of a 105 week 544 household subsampled dataset - exact same dataset as used for the 4_6_b_forecast_NN_hh.ipynb model

Code based on link below

see https://machinelearningmastery.com/how-to-develop-convolutional-neural-networks-for-multi-step-time-series-forecasting/

4_6_a_forecast_NN_daily.ipynb

Pytorch 7 day forecast mixed input embedded categorical model for daily data

4_6_b_forecast_NN_hh.ipynb

Pytorch 7 day forecast mixed input embedded categorical model for half hourly data

More testing experimentation with features (and NN parameters) may have improved the model, but only had sufficient time for a couple of runs for this model.

Forecast prediction is for the last week of a 105 week 544 household subsampled dataset - exact same dataset as used for the 4_4_b_forecast_GB_hh_ds.ipynb

4_7_forecast_multivariate_encoder_decoder_LSTM.py

python script in scripts folder.

Encoder-Decoder LSTM uses each of the time series variables to predict the next 7 days energy consumption(each half hour).

Each one-dimensional time series provided to the model as a separate sequence of input.

LSTM creates an internal representation of each input sequence that will together be interpreted by the decoder.

Code based on link below

See https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/

Conclusions

See Results_and_Conclusions.ipynb for more details