Project design

The goal of the project is to predict the hourly number of requests of a web server one day in advance.
The input is a time-series of hourly requests and the models implemented are a seasonal state-space ARIMA model and a sequence to sequence long short term memory neural network.

Tools

The project has been coded in Python 3.6, the data for the analysis, as well as the results have been stored in pandas DataFrame objects.

The algorithm and metrics used for modeling the seasonal arima model belong to statsmodel library. The evaluation metrics belong to the sklearn library.
The neural network have been implemented using the keras library.
The output has been loaded into csv files and files representing the serialization of models.
The visualizations have been realized using matplotlib and seaborn.

Data

The dataset for this project is seven month's worth of all HTTP requests to the University of Saskatchewan's WWW server. (http://ita.ee.lbl.gov/html/contrib/Sask-HTTP.html)
The logs are an ASCII file with one line per request, with the following columns.

Variable Name	Type	Description
remotehost	String	Remote hostname (or IP number if DNS hostname is not available).
rfc931	String	Remote logname of the user.
date	Datetime	Date and time of the request
request	String	The request from the browser or client.
status	String	The HTTP status code the server sent back to the client
bytes	integer	The number of bytes (Content-Length) transferred to the client.

Algorithm(s)

First a time series of the hourly requests have been produced, then the time series of the hourly requests has been analyzed to detect seasonality, lags relevant for autocorrelation, partial autocorrelation and stationarity. Those preliminary results have been used to create the parameter space on which to run a grid search with the goal of minimizing mean square error between the real values and the predicted ones.
The resulting SARIMAX (1,1,1,0,1,2,24) produce the best 24 hour forecast in term of MSE between real values and predicted ones.

The second model is a sequence to sequence deep neural network with two LSTM layers, one acting as an encoder, the other acting as a decoder. The encoder state is passed to the decoder in a ‘teacher forcing’ way.