Exercise Sheet 1: Linear Models

December 1, 2022

The overarching topic of our exercises will be Demand Forecasting, i.e., time series predictions for different products in different locations, for example of a grocery chain.
The data sets to be used for the exercises can be found here:
https://github.com/FelixWick/understanding_ml/tree/main/exercises
The description of the data sets and variables therein can be found in the file `datasets_description.txt` in the same directory.

1) Familiarize yourself with the data and conduct an Explanatory Data Analysis (EDA), using summary statistics, visualizations, etc.
2) Univariate model
   a) Predict the demand (expressed by sales values) of all product-location-date combinations in `test.gzip`, using a univariate time series model of your choice (without using the actual values in `test_results.gzip`), e.g., Exponential Weighted Moving Averages (EWMA, included in python package *pandas*, hint: You can include additional columns in the group-by for the EWMA estimation.). This static train-test setup corresponds to multi-step forecasting with a mix of forecast horizons from one day to six months.
   b) Predict now with a single forecast horizon of one day, corresponding to a dynamic setup with a sliding window over the test data set. You need to use the sales information from `test_results.gzip` for this. But make sure not to use the sales information of the day to predict (future information at prediction time).
3) Evaluation of predictions
   a) Compute the mean absolute deviation (MAD) and the mean squared error (MSE) of your predictions on `test.csv` compared to the respective actual values in `test_results.gzip`. (For this, you need to merge the two data sets by means of the product-location-date keys.)
   b) Compute MAD and MSE only for product-location-date combinations with promotions as well as time windows of one week before and one week after events.
   c) Plot the time series of your predictions and actuals (each summed up over the different product-location combinations) for the time period of `test.gzip`.
   d) Repeat this for each location (sums only running over products) and product-group level 3 (sums running over locations and products in respective product group).
4) Linear regression
   a) Repeat the predictions from exercise 2 with a linear ML model (i.e., training all product-location time series together in a multivariate way), e.g., linear regression from the python package *scikit-learn*. This requires an i.i.d. (identical and independent distributed random variables) assumption over the different time steps (and products and locations). You need a one-hot encoding for the categorical variables to include several products and locations in one model. You can choose one of the two setups in exercise 2) a and b. Repeat the evaluations with this model.
   b) Compare the importance of the different features by means of the learned model parameters.
5) Use predictions of your univariate method from exercise 2 (also applied on `train.csv`) as additional features (or replacing one-hot encoded product and location features) for your linear regression model from exercise 4. Again, you can choose one of the two setups in exercise 2) a and b (or a mix of the two). Repeat the evaluations with this model.
6) Turn your linear regression into a multiplicative model by using an appropriate link function, like in Poisson regression from *scikit-learn*. Again, you can choose one of the two setups in exercise 2) a and b. Repeat the evaluations with this model.