Compulsory assignment 2 **Submission deadline: 03.11.2023**

This compulsory assignment consists of 4 exercises. Exercises must be solved in groups (assigned in Canvas) and uploaded to Canvas by the submission deadline. Please provide a single `.html` or `.pdf`-file, entitled `dat320_comp2_groupX.html` (or `.pdf`), where `X` should be replaced by your group number (1-16). The file should be structured into sections (one section per exercise) and subsections (one subsection per task). The usage of R markdown (R package `knitr`) is strongly recommended.

Exercise 1 (Univariate Forecasting)

The dataset `austres` from package `datasets` contains a quarterly time series representing the number of Australian residents from 1971 to 1993. The goal of this exercise is to forecast the number of residents and explore how ARIMA hyperparameter selection is performed.

Material provided:

- `austres` dataset from package `datasets`

Tasks:

(a) *Load and plot the dataset along with the ACF and PACF. What can we say about stationarity, trend and seasonality? Which ARIMA model (model hyperparameters) would you suggest based on the exploratory analysis (no model training needed, yet)?*

(b) *Implement a function to select the hyperparameters of an ARIMA model (i.e., your own `auto.arima` function). Use the description from the software paper*

https://www.jstatsoft.org/article/view/v027i03/v27i03.pdf, Section 3.2.

Information can also be found in

https://otexts.com/fpp2/arima-r.html.[a]

Include an option to use either `AIC` or `BIC` as selection criterion. You may make the following simplifications:

- *ignore all seasonal terms, instead only use AR and MA terms, along with differencing;*
- *set an upper limit of $k < 5$ (number of AR terms) and $q < 5$ (number of MA terms), and $d \leq 3$;*
- *always use a model with intercept, if $d \leq 1$, and a model without intercept otherwise. You may ignore additional instructions about intercepts (constants) in the literature.*

You may use the function `Arima()` from package `forecast` to train the parameters for a given set of hyperparameters. Note that the function `auto.arima()` from package `forecast` may suggest slightly different parameters due to the simplifications suggested.

(c) *Divide the dataset into train and test set, using 70 data points for training and 19 for testing. Use your implemented function to select the model hyperparameters. Use*

AIC and BIC as criteria for training, and compare both model parameters. Further, compare with the hyperparameters suggested in task a. Note: If you are not successful with task b, use the function `auto.arima` from package `forecast` instead.

(d) *Compute forecasts with prediction horizon $h = 19$ from each of the models trained in task c (hyperparameter selection using BIC, hyperparameter selection using AIC, as well as your own hyperparameters from task a). Compare with the ground truth on the test set using RMSE. Further, train the four baseline models (average, drift, naive and seasonal naive) and evaluate them on the test set. Provide a table showing the test set performances of all methods. Which model performs best?*

[a] Note that the variable p in the book is used to indicate the number of AR terms, while we use k in the lecture, instead.

Exercise 2 (Univariate Forecasting with Seasonality)

For this exercise, we use the dataset *co2.csv* that contains information about CO2 emission around the world, measured as daily samples between 2019 and 2023 in different regions. The dataset contains a column `date`, a column `co2` representing the measured emission level, and a categorical column `region`. Note that this dataset is different from the CO2 dataset from exercise 4 on assignment 1.

Material provided:

- `co2.csv`: dataset

Tasks:

(a) *Load the dataset `co2.csv`. Sum over all regions in order to obtain total CO2 emission per day. Plot the time series along with its ACF and PACF and investigate its properties.*

(b) *Implement a cross-validation pipeline for the dataset using 23 weeks as an initial fold for training, and a 21-days-ahead forecast (3 weeks) in each fold. The pipeline should be applicable for an arbitrary model type.*

(c) *Use the cross-validation function implemented in task b to train 4 models:*

- *an ETS model with trend and additive seasonality*
- *an ETS model with trend and multiplicative seasonality*
- *an ARIMA model with order $(k = 5, d = 1, q = 2)$*
- *a SARIMA model with order $(k = 2, d = 0, q = 2)(K = 0, D = 1, Q = 2)$.*

For all models with seasonal terms, use weekly seasonal periods. Compute the mean and standard deviation on the test RMSE across the folds for each model. Which model works best? Discuss the differences between the models.

(d) *For each of the models trained in task c, use the last cross-validation fold to investigate the model parameters, as well as the residuals. Plot the predictions on the last fold.*

Exercise 3 (Univariate Forecasting with exogeneous variables)

The dataset *watershed* contains measurements of the gauged runoff level of a river in North America (`gauge`) along with climate data like temperature (`temp`) or rainfall (`rain`). The goal is to model the relation between the climate variables and the gauge variable.

Material provided:

- `watershed.csv`: dataset

Tasks:

(a) *Load the dataset and change the date column to an appropriate date format. Perform an exploratory data analysis and plot the variables `gauge`, `rain` and `temp`. Evaluate whether the time series `gauge`, `rain` and `temp` are stationary.*

(b) *Investigate the relationships between the given variables using ACF, PACF, and CCF. What do the time series have in common? Further, apply a Granger causality test with gauge as target variable and either rain, or temp, or both as predictors. Interpret the test results!*

(c) *Split the dataset into a train dataset (containing 30 hydrological years) and a test dataset. Train the following models to predict the gauge variable:*

- *a linear regression model with predictor `rain`,*
- *a distributed lag model with predictor `rain`,*
- *an ARIMA model,*
- *a dynamic regression model on predictor `rain`,*
- *a dynamic regression model with lagged predictors on predictor `rain`.*

Provide a table comparing the performance metrics RMSE and R2 of each model on the test set.

(d) *Which model performs best on the test set? Which model is most suitable for a practical problem setup, where the influence of rainfall events on the gauged runoff of rivers should be modeled?*

Exercise 4 (Linear models with auto-regressive errors)

In this exercise, we will use our knowledge about auto-regressive models (AR) to train an ordinary linear regression model with auto-correlations in the model errors. Assume a linear regression model of the following type:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

The model fulfills the model assumption of identically distributed errors $\boldsymbol{\varepsilon}$ following a Gaussian distribution with mean 0 and variance σ^2. However, the model violates the independence assumption, i.e. ε_i is not independent of ε_j for $i \neq j$.

Tasks:

(a) *First, assume that the model does not have an intercept. We can try to model the errors ε_i with an $AR(1)$ model, i.e. for all samples i*

$$y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \varepsilon_i, \tag{1}$$

$$\varepsilon_i = \varphi_1 \varepsilon_{i-1} + \eta_i, \tag{2}$$

$$\eta_i \underset{iid}{\sim} N(0, \sigma^2). \tag{3}$$

Note that $\boldsymbol{x}_i$ denotes the i-th row of $\boldsymbol{X}$ (as a column vector). In order to train the parameters $\boldsymbol{\beta}$ and φ_1 of such a model, we use three steps:

a) Train the parameters $\hat{\boldsymbol{\beta}}$ using ordinary least squares (without taking the violation of the assumption into account).

b) Estimate the AR parameter φ_1 from the residuals

c) Re-estimate the parameters $\hat{\boldsymbol{\beta}}$ using ordinary least squares from the transformed data $\tilde{y}_i = y_i - \varphi_1 y_{i-1}$ and $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \varphi_1 \boldsymbol{x}_{i-1}$, respectively, for $i > 1$.

Examine the model formulas and explain why applying the transformation in step c) is a valid step to estimate parameters in the model above (under the assumption that φ_1 is correctly estimated)! Hint: it is key to argue that a regression model on the transformed predictors $\tilde{\boldsymbol{x}}_i$ and target variable $\tilde{y}_i$ has uncorrelated errors. Why is that?

(b) *Investigate the following questions:*

- *Is the parameter vector $\boldsymbol{\beta}$ the same in the linear regression model on the transformed predictors as in the linear regression model trained on the original predictors?*
- *Is the same concept applicable for general $AR(k)$ error terms?*
- *Can you think of any practical limitations associated with the described procedure?*

(c) *Is the same concept applicable if the model contains an intercept? If yes, does the relation between the parameters $\boldsymbol{\beta}$ in the transformed and the original space still hold if we have a model with intercept? Justify based on the model formulas!*

(d) *Implement the procedure in R for $AR(1)$ models. Try the model with your implemented parameter estimation procedure out on the dataset from exercise 3! Are the model residuals uncorrelated after applying the transformation?*