Professorship of Operations Management // Center for Digital Transformation
TUM School of Management
Technical University of Munich

Business Forecasting: Project 1 (Programming)

Large retailers typically offer a wide assortment of products, ranging from fast-moving consumer goods to niche or seasonal items. This diversity creates substantial complexity in demand forecasting, as sales patterns often vary significantly between product categories. Some time series may exhibit strong seasonal trends, while others might be irregular, sparse, or influenced by external factors such as promotions or supply chain disruptions. Consequently, the predictability of demand across product classes is often inconsistent, making a one-size-fits-all forecasting approach ineffective.

To address this challenge, forecasting methods must be carefully selected and tailored to the characteristics of each individual time series. This makes the design of a dynamic and flexible forecasting system not only desirable but necessary for practical applications in retail analytics.

In this project, we work with real-world time series data provided by an international retail company. The dataset contains aggregated sales figures at the product class level. These values represent the number of units sold over time, and in cases where items are sold by weight (e.g., in kilograms or liters), fractional values may appear in the data.

Task 1

Your task is to implement a forecasting system as Python script (i.e. a `.py` file). It should be able to forecast up to 5 time series and dynamically select suitable models for each time series according to some logic. You as expert team are tasked with devising a simple logic.

At the top of your script, you should allow your users to set some variables as input:

- `file_name`: `str`, file name of data to be read in.
- `ids_to_forecast`: `list[str]`, time series names of time series selected to be forecasted.
- `metric`: `str`, either `'mape'` or `'mse'`.

If the input for `metric` is not correct, throw an error and stop the execution.

Your forecasting system then should do the following:

a) Read in the data as given in `proj1_exampleinput.csv`.
b) Perform any necessary data preparation and filter the data such that you obtain the specified time series.
c) Select the models for forecasting using the following logic. For each category - if not explicitly specified differently - use 2-4 suitable models from the lecture:
   - Category 1: Time series is shorter than two seasonality periods (pick only one model).
   - Category 2: Else, time series contains zeroes and has high seasonality.
   - Category 3: Else, time series has high seasonality and only positive values.
   - Category 4: Times series has low or no seasonality and zeroes.
   - Category 5: All remaining.
   - Additionally, in all cases always run the Naive forecast as benchmark.
d) Perform a cross validation simulating forecasts for one year, every half year for an evaluation period of 2 years. Do not perform cross-validation for time series shorter than 48 observations.
e) Calculate the accuracy for all models over all errors with the metric chosen by the user. If at least one value of a time series is zeroes, throw a warning if MAPE is used.
f) The system should now select the best performing single forecasting method per time series according to the metric.

g) Now forecast into the future and only select the chosen forecast method per time series. Try to make this as efficient as possible.
h) Create the following outputs as print for each time series: Name of the time series, chosen model, accuracy metric and accuracy value of the chosen forecast method and the Naive as benchmark.
i) Save the forecasts as `.csv` file in the directory.

Remarks

- The input variable names at the top of the script have to be named exactly as in the task description.
- You can assume that the input data always has the same format and is stored in the same directory as your Python file.
- Your code must be commented explaining what you are doing, but most importantly at every decision point why you have chosen to implement it in a certain way.
- You should mostly use the Python libraries provided in the extended enviroment at the beginning of the lecture. If you need additional libraries, they must be available in the Python Package Index (pypi.org). Explain in a comment, why you need them.
- Take care and extra thought, when selecting the forecasting methods for each case. Consider which methods are suitable, but also give you the most flexibility. Give the reasoning for your selection.

Task 2

This exercise has to be submitted individually. If you submit Task 1 as a **group**, please select time series **A** or **B**. If you are submitting alone, please take time series **A**.

Time series to analyse and explain the accuracy:

- **A**: C1038, C2716, C6022
- **B**: C1096, C3029, C6810

You are now tasked with explaining to other data scientist in your company why for the 3 time series we have higher or lower accuracy and why certain models have been selected. Your task is to create a **precise and compact report (2-3 pages)**. You should support your claims and findings with additional data analysis.

The report should be handed in as `.pdf` file. You may for example use Markdown or Word to create it.

Explain at least the following for each time series:

- What was the achieved accuracy and which model was selected?
- What properties of the time series have contributed to the chosen best-fit model working well?
- Does the model properly model the patterns in the data or are there patterns that it does not describe?
- Do you think that out of all the models in the lecture so far, there would be another model that might perform better than the chosen one?

Remarks

- This report is for other data scientists. So you may and should use the scientific terms and can suppose a certain level of statistical understanding from the readers.
- Feel free to support your analysis with pictures and plots, but make sure they are relevant for your findings.
- Your analysis should be deeper than your report, so you should condense it to highlight the relevant points.

Submission

- The full project has to be sumbmitted in the Moodle course under section `Project 1` latest until 16.06.2025 at 11:30 p.m.
- Every student has to submit both Task 1 and Task 2 individually. For registered groups, Task 1 can be the same file.
- Submission for Task 1: One single `.py` file with necessary comments (see Task 1 Remarks) included.
- Submission for Task 2: One PDF file.
- For your submission to be graded, you must be registered in tumonline for the exam.

Group registration

You are allowed to work in groups of up to 2 people for Task 1. If you decide to so, please send a mail to dominik.proschmann@tum.de with your partner in cc. In this mail, mention who does part A and B for Task 2. Remember that only communication using your official `@tum.de` email adress is allowed.