Project 2 - Forecasting Models with LamaH dataset

Leon Beccard 12133103 e12133103@student.tuwien.ac.at
Jonas Unruh 12331457 e12331457@student.tuwien.ac.at

05.01.2024

1 Introduction

Weather forecasting models are highly demanded tools in the era of climate change and machine learning. Further, machine learning models (MLMs) forecasting future time-series events such as weather or different climate behaviours have become quite important. In terms of resource efficiency, MLMs can immensely reduce resource usage compared to traditional forecasting models. Also the general speed and efficiency of MLMs forecasts usually excel traditional models. Another important topic is feature extraction, which MLMs can automatically extract from raw weather data. MLMs can be trained to recognize patterns that precede extreme weather events, such as hurricanes, tornadoes, or heatwaves. By identifying subtle indicators in the data, these models can help in early detection and warnings. In conclusion, to predict complex time-series future behaviours involves employing sophisticated forecasting model techniques.

2 Background

To predict the next days precipitation given on a dataset with several hydrological informations, several decisions are required to reach a successful and reliable forecasting model. Several different types of models have been developed to forecast time series events, where one needs to be chosen. On the one side statistical models such as time series analysis leverage patterns and trends present in historical data to make predictions. These models for example can examine how past precipitation values relate to each other and use this information to forecast future values. On the other side machine learning models like decision trees, random forests, support vector machines, or neural networks learn complex patterns to predict future precipitation. These models can capture nonlinear relationships between various meteorological factors and precipitation, allowing for more intricate predictions. Weather forecasting involves processing immense amounts of data, including atmospheric pressure, temperature, humidity, wind speed and more. Therefore, preprocessing data quality plays an important role to provide reliable data.

3 Data Analysis

3.1 Introduction to the Data

The data we are using is part of the LamaH (Large-Sample Data for Hydrology and Environmental Sciences) dataset. This dataset contains hydrological data gathered through different sensors spread along rivers over an area of 170.000km^2 in Central Europe.

These sensors have gather a wide variety of data across a time period of about 40 years. The data attributes include values such as precipitation and temperature and therefore widely differ at each measurement point, mostly due to the massive geographical differences present in Central Europe.

The dataset its self is freely available at `https://doi.org/10.5281/zenodo.4525244`. [Kli]

For this exercise we focus on a subset of the upstream basins timeseries. From these 859 files we randomly select 100 as our dataset.

3.2 Descriptive Statistics

For the descriptive statistics we decided to have a look at the distribution of the data for every feature compared over all gauges. This left us with on plot that mostly showed the vast difference in value ranges across the different features. To get around this issue all features were split up into groups and plotted. This resulted in more plots but each with visible data. It became very visible that there were quite massive differences in the values for every feature, which is most likely to be attributed to geographical factors associated with each location. Figure 1 highlights this issue. The lowest temperature present in the data is somewhere around -38°C, while the max lies around 37°C. This represents a difference of about 75°C. Similar differences were visible in the other plots. The obvious conclusion from this is that the different time series cannot be compared as is. They need some sort of value that gives reference to its geographical location.

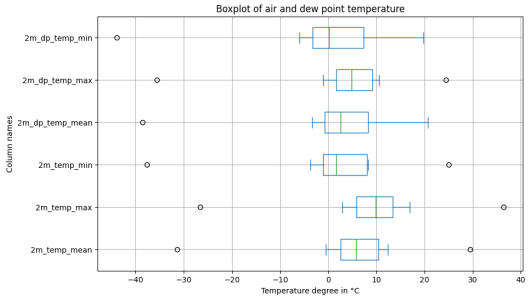

Figure 1: Distribution of temperature values

We did not conclude that any outliers were present in the data. Even though some unbelievable values were present in the data, these numbers could be confirmed by news articles and the surrounding values.

The data itself is mostly normally distributed (Figure 2). The values year, month, day and day of year are all evenly distributed as they should be. The values that are heavily skewed look like this because they have quite a few data points with a value of 0. These are all correct though, as they correspond to features such as precipitation which are not always present.

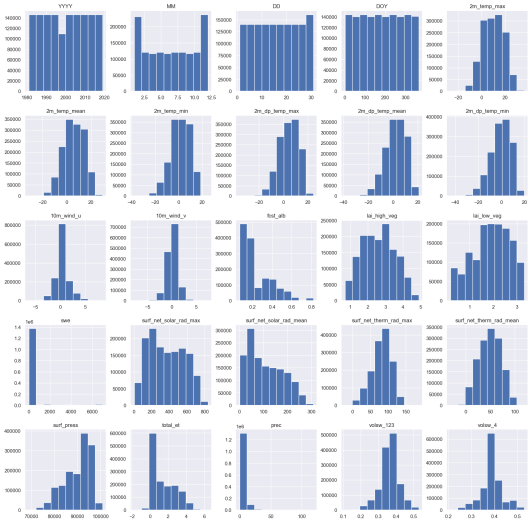

Figure 2: Distribution of feature values

3.3 Data Exploration

The heatmap (figure 3) shows the correlation of every feature to one another. It clearly shows some very high correlations, that are mostly within the predefined groups such as temperature within itself or obviously related such as evaporation and temperature. There were no obvious weird values within this figure.

To find seasonality in the target value, which would be relevant for some time series models, a plot created that shows the average monthly precipitation over the entire timeline (figure 4). The red vertical lines are the start date for a new year in this plot. There seems to be a spike in the summer for most years.

3.4 Data Cleaning and Preprocessing

The data preprocessing itself was quite straightforward as the data is very clean already. To ensure it works with all models and only minimal processing would be necessary before each training, these steps were followed:

- add area attributes
- setup next day prediction
- create full data frame

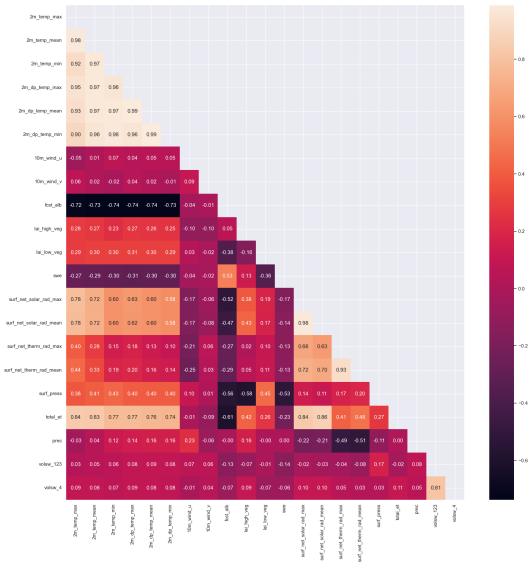

Figure 3: Heatmap for feature correlation

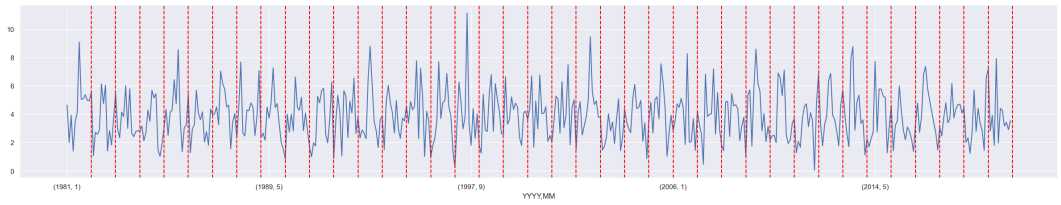

Figure 4: Timeline of average monthly precipitation of the gauge with ID 158

- split training and test data
- normalization

Add area attributes:

As our time series need some sort of geographical identifier to be comparable, we took the relevant elevation, longitude and latitude values from the gauges file for each of our time series.

Setup next day prediction:

As the goal of our models is to predict next days precipitation based on today's values, we decided to shift every data frame by one row. This means we dropped the last row of every data frame and shifted our target values down one.

Create full data frame:

To have one data frame that would be usable for each model we concatenated all data frames into one large one.

Split training and test data:

As defined in the exercise description we applied a 70, 30 split on the data frame, creating a train and test data frame.

Normalization:

The features have a very large difference in value ranges between them. That is why we decided to apply normalization to the dataset.

Sadly all of this would only work for three of the four chosen models. The multi-series Autoregressive Model needed some different preprocessing that was applied directly before training. This preprocessing only consisted of dropping all features except the target values and adding those together to a data frame with a datetime index. This was the case because this model treats every time series as an independent series, which is very sensible, but also suppresses us from adding any values that solely reference one of the series.

4 Experiments and Results

4.1 Experiments

The models performance will be evaluated on its next day precipitation prediction capabilities. This means the features for the current day are given. Our performance metric will be the Root Mean Squared Error (RMSE).

We will also compare feature importance across models to see which features actually contribute the most to each model. To guarantee comparability, the feature importance will be measured using permutation.

Regression Model:

- LinearRegressor
- fit_intercept: True

Random Forest Model:

- n_estimators: 180
- min_samples_split: 2
- min_samples_leaf: 4
- max_features: 'sqrt'
- max_depth: 70

MLP Model:

- MLPRegressor
- max_iter: 100
- activation: 'relu'
- solver: 'adam'

Autoregressive Model:

- lag: 24
- regressor: Ridge

4.2 Results

4.2.1 RMSE

The RMSE values presented in figure 5 show the performance of each model. The values seem to be quite close together, especially the regression model and the MLP model have almost the same RMSE. This was quite surprising as the MLP model is far more complex, but apparently in its configuration it cannot fully deal with the data.

The random forest model performed a little bit better, which is not a huge surprise, as this type of model usually performs quite well on most data. Still the result is "only" a 9% increase in performance when compared to the MLP model.

The best performing model was the autoregressive model, which is quite surprising, especially seeing as it is arguably the simplest model of all, as it only takes the precipitation of previous days into account. Then again it is also the only model that actually looks at the data as a time series, which seems to be quite relevant for performance. It can show a 9.7% performance increase in comparison to the random forest model and a 22.3% increase when compared to the regression model. These are quite large improvements especially when considering the training time. The random forest and MLP model took approximately 30 minutes to train. On the contrary to the autoregressive model, which was trainable in a time similar to that of the regression model, which were just a few seconds.

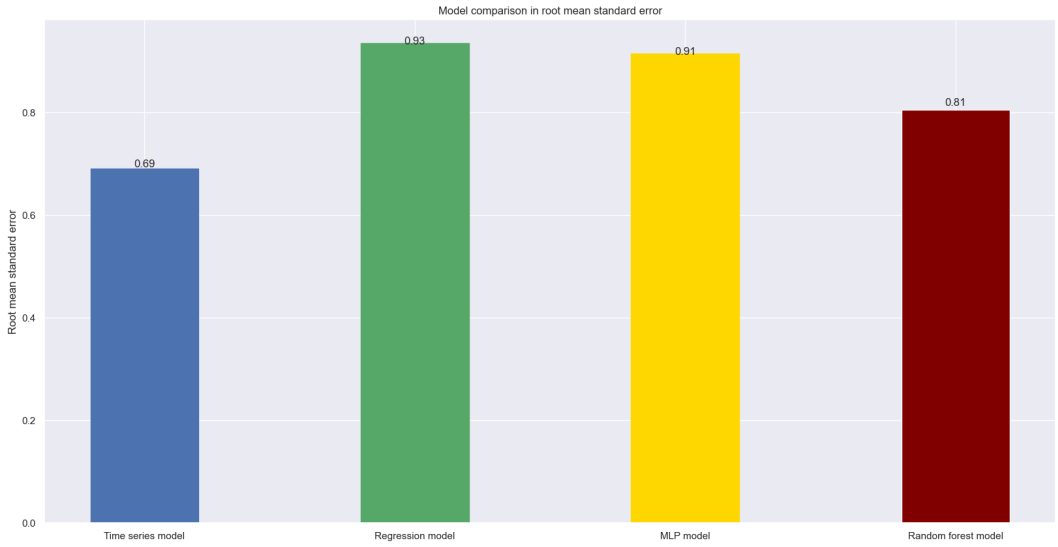

Figure 5: RMSE of every model

4.2.2 Feature Importance

The feature importance is quite interesting. Figure 6 shows the feature importance for the regression, random forest and MLP models. But basically only the values for the MM and

DOY features for the regression model are visible. Figure 7 excludes these two features, which leads to a clearer picture in regards to the importance of each feature to every model.

Interestingly enough the random forest and MLP model have the exact same feature importances. We believe this to be an interesting coincidence and not something that would repeat itself in every trained model. Nevertheless these features seem to be relevant regardless of model, which could be something to consider and keep in mind when further optimizing the workflow.

The regression model has some very interesting feature importances. Most likely this is due to the way the model calculates the actual prediction. Removing those features does leave us with similar results as the other models as well. The importances are different, but no highly relevant feature is completely left out. In general it seems that changes in the data affect the regression model more than our other two models.

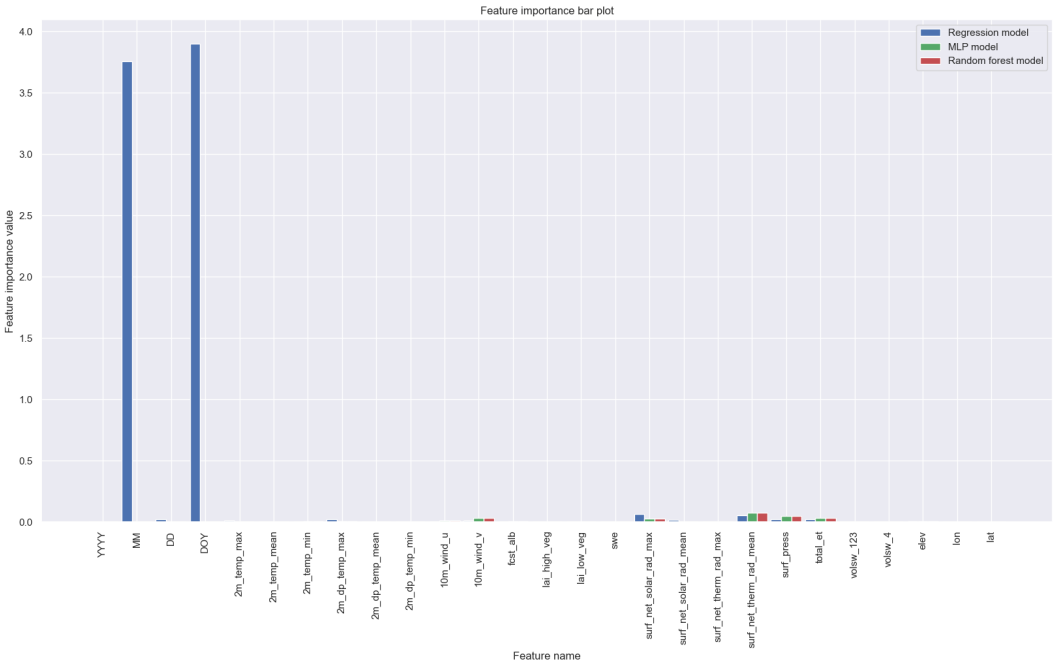

Figure 6: Feature importance for every model and with each attribute

5 Conclusions

Several observations have been found. As stated in the beginning time series models differ a lot to machine learning models such as linear regression, neural networks or random forests. This leads to a more difficult task to create a time series model using every value of the dataset instead of only using the precipitation values. This task should be considered for future work. The difference between the two types of models also makes it hard to compare the feature importance since we can only compare the lags of the time series models. Another future task is hyperparameter tuning, which has taken a lot of computational resources and

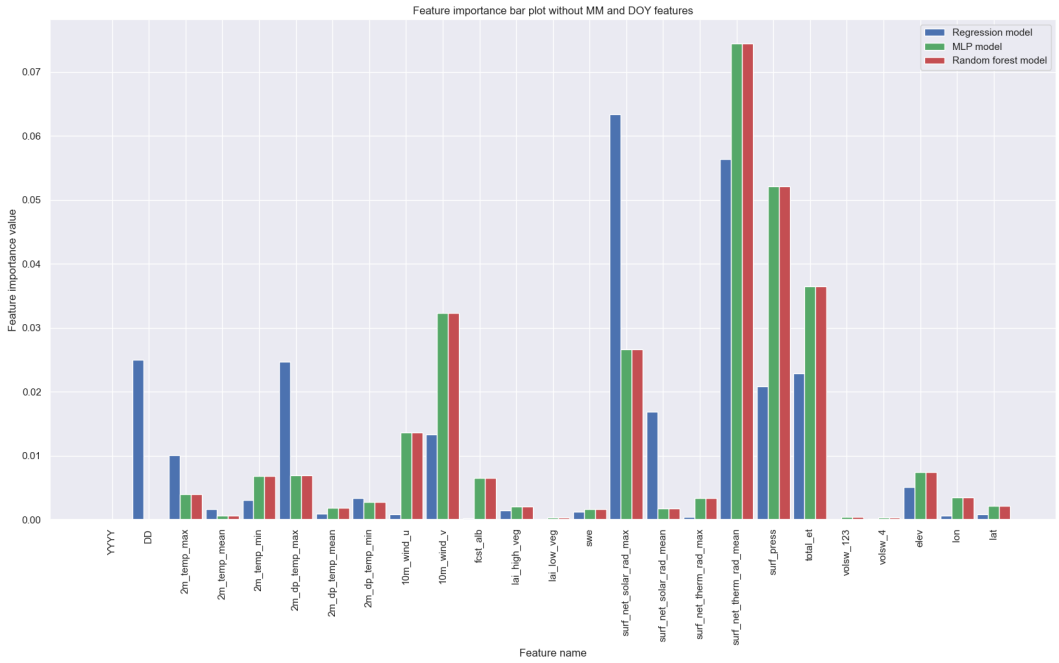

Figure 7: Feature importance for every model excluding MM and and DOY

time. Since the randomized grid search did not lead to better results a complete grid search should be taken into account next. For the time series model, hyperparameter tuning should be taken into account as well as more complex models such as the SARIMA model since the dataset has a seasonal trend. For hyperparameter tuning a grid search, random search and a bayesian search can be performed on the regressor parameters and the amount of lags. This leads to another task for future work. Even though the dataset is quite comprehensive, it is important to note that forecasting models may encounter limitations due to unaccounted variables in the dataset.

References

[Kli] Klingler, C., Schulz, K., and Herrnegger, M. LamaH-CE: LArge-SaMple DAta for Hydrology and Environmental Sciences for Central Europe.