main

March 3, 2025

1 Introduction

Time series modeling is a hot topic in tech right now. Bridging the worlds of statistical modeling and machine learning, time series modeling allows one to model processes based on one thing: time. For the majority of the discipline's existence, stochastic models have dominated the methodology. Primarily, these include ARIMA and SARIMA models. While robust to many troublesome patterns found within your run of the mill time series, they can't solve everything. Parametric models, while accurate, always suffer from violated assumptions. One issue these models struggle with is non-constant variance, which was solved by Robert F. Engle's ARCH model [1].

New innovations are still being made to solve the holes in these models. A 2017 research paper written by Hristos Tyralis and Georgia Papacharalampous [2] suggests that the use of random forest for endogenous variable selection in multivariate time series produces better forecasts than their univariate counterparts.

In this project, we examine the BCG Jena Weather Station Dataset from 2017-2024 [3]. Utilizing the methodology of Tyralis and Papacharalampous, we investigate the effectiveness of random forest-engineered time series models on weather in comparison to other contemporary methods.

2 Data Processing

[6]:
```
import kagglehub
import os
import pandas as pd
import matplotlib.pyplot as plt

path = kagglehub.dataset_download("matthewjansen/
 ↪bgc-jena-weather-station-dataset-20172024")
files = os.listdir(path)
csv_file = [file for file in files if file.endswith('.csv')]
dataset_path = os.path.join(path, csv_file[0])
df = pd.read_csv(dataset_path)
print(df)
```

```
                 Date Time  p (mbar)  T (degC)  Tpot (K)  Tdew (degC)  \
0       2017-01-01 00:10:00    999.77     -4.91    268.27        -8.41
1       2017-01-01 00:20:00    999.63     -5.05    268.13        -8.37
2       2017-01-01 00:30:00    999.54     -4.98    268.21        -8.38
```

```
3       2017-01-01 00:40:00    999.40     -4.88    268.33        -8.56
4       2017-01-01 00:50:00    999.17     -5.17    268.06        -8.74
…                       …         …        …        …             …
420777  2024-12-31 23:20:00    997.74     -0.78    272.55        -2.64
420778  2024-12-31 23:30:00    997.81     -1.42    271.90        -2.71
420779  2024-12-31 23:40:00    997.88     -1.41    271.90        -2.80
420780  2024-12-31 23:50:00    997.92     -1.63    271.68        -2.96
420781  2025-01-01 00:00:00    997.51     -1.54    271.81        -3.00

        rh (%)  VPmax (mbar)  VPact (mbar)  VPdef (mbar)  sh (g/kg)  …  \
0         76.3          4.24          3.23          1.00       2.01  …
1         77.4          4.19          3.24          0.95       2.02  …
2         76.9          4.21          3.24          0.97       2.02  …
3         75.2          4.25          3.19          1.05       1.99  …
4         75.8          4.15          3.15          1.01       1.96  …
…          …             …             …             …          …  …
420777    87.1          5.77          5.03          0.74       3.14  …
420778    90.9          5.50          5.00          0.50       3.12  …
420779    90.2          5.51          4.97          0.54       3.10  …
420780    90.6          5.42          4.91          0.51       3.07  …
420781    89.7          5.46          4.89          0.56       3.06  …

        wv (m/s)  max. wv (m/s)  wd (deg)  rain (mm)  raining (s)  \
0           0.78           1.56     184.0        0.0          0.0
1           1.52           1.92     202.6        0.0          0.0
2           0.98           1.78     227.4        0.0          0.0
3           1.16           1.80     212.5        0.0          0.0
4           1.50           2.64     222.1        0.0          0.0
…            …              …         …          …            …
420777      1.27           3.48     291.4        0.0          0.0
420778      1.02           2.39     297.4        0.0          0.0
420779      1.08           1.91     315.8        0.0          0.0
420780      1.48           2.94     208.6        0.0          0.0
420781      1.95           3.80     140.5        0.0          0.0

        SWDR (W/m²)  PAR (µmol/m²/s)  max. PAR (µmol/m²/s)  Tlog (degC)  \
0               0.0              0.0                   0.0         7.10
1               0.0              0.0                   0.0         7.72
2               0.0              0.0                   0.0         8.77
3               0.0              0.0                   0.0         9.36
4               0.0              0.0                   0.0         9.45
…                …                …                     …           …
420777          0.0              0.0                   0.0         6.27
420778          0.0              0.0                   0.0         6.21
420779          0.0              0.0                   0.0         6.20
420780          0.0              0.0                   0.0         6.20
420781          0.0              0.0                   0.0         6.21
```

```
        CO2 (ppm)
0           434.3
1           434.1
2           430.4
3           430.6
4           429.5
…           …
420777      453.5
420778      453.7
420779      453.8
420780      453.4
420781      457.5

[420782 rows x 22 columns]
```

The BCG Jena Weather Station Dataset contains 22 measurements of weather related processes such as pressure, temperature, and namely, rainfall. For the sake of this project, we will be focussing on the rain measurements. In this particular dataset, rainfall is measured in 10 minute intervals. While informative, this scale doesn't present meaningful insights. Thus, we will convert the dataset to a day-by-day amount.

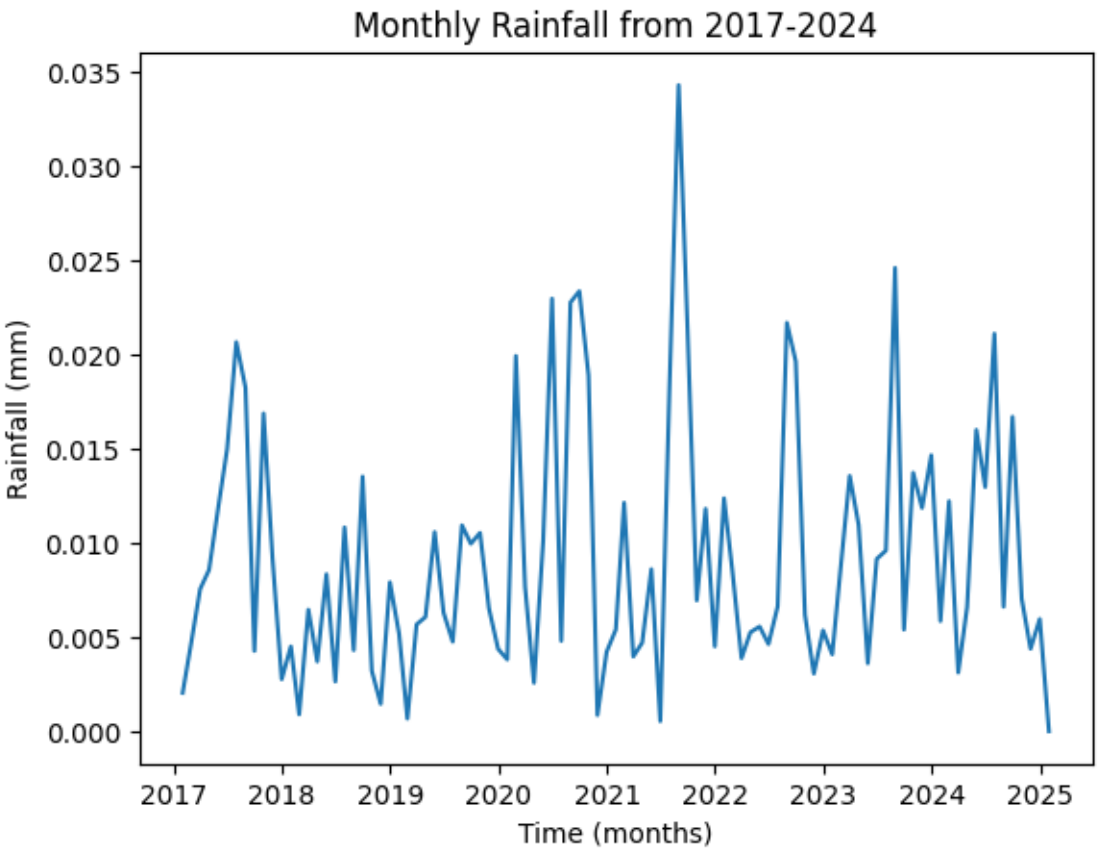
[7]:
```
import pandas as pd

rain = df[["Date Time", "rain (mm)"]]
rain["datetime"] = pd.to_datetime(rain["Date Time"])
rain.set_index("datetime", inplace=True)
daily_means = rain["rain (mm)"].resample("ME").mean()

plt.plot(daily_means)
plt.xlabel("Time (months)")
plt.ylabel("Rainfall (mm)")
plt.title("Monthly Rainfall from 2017-2024")
plt.show()
```

```
/var/folders/94/m6gj35yn3zddpm3bq3jz8w0h0000gn/T/ipykernel_8397/2196414821.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  rain["datetime"] = pd.to_datetime(rain["Date Time"])
```



3 Feature Engineering

We will first proceed by creating endogenous variables for the random forest model.

[13]:
```
hourly_means = rain["rain (mm)"].resample("h").mean()
daily_means = rain["rain (mm)"].resample("D").mean()
monthly_means = rain["rain (mm)"].resample("ME").mean()
yearly_means = rain["rain (mm)"].resample("YE").mean()
```

4 References

1. Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica, 50(4), 987–1007. The Econometric Society. https://doi.org/10.2307/1912773.
2. Tyralis, H., & Papacharalampous, G. (2017). Variable Selection in Time Series Forecasting Using Random Forests. Algorithms, 10(4), 114. https://doi.org/10.3390/a10040114
3. Jansen, M. (2024). BGC Jena weather station dataset (2017–2024) [Dataset]. Kaggle. https://www.kaggle.com/datasets/matthewjansen/bgc-jena-weather-station-dataset-

20172024.