Wolt Test Assignment Report: Predicting number of couriers

MIKHAIL SILAEV

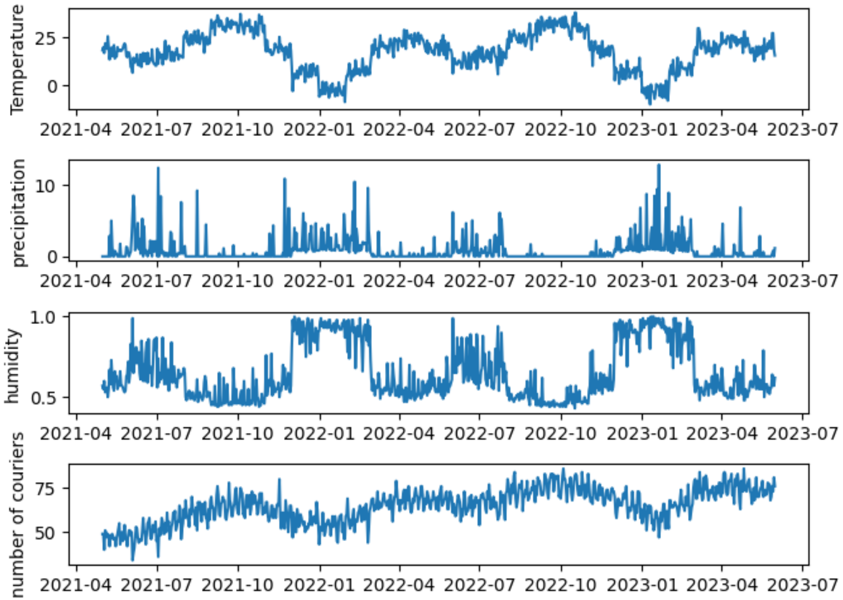
2025-01-21

Introduction

▶The problem of predicting time series is crucial for the future activity planning. The dataset given contains daily courier number combined with the weather data.

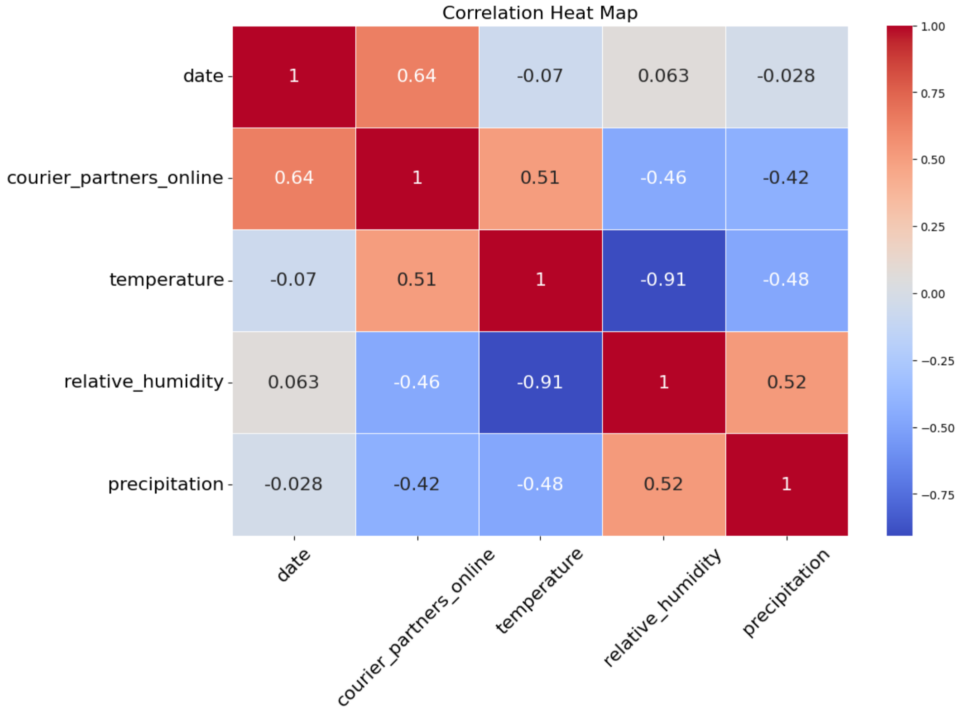
▶Outlier-cleaned time series on this slide show visible correlations of “number of couriers” with temperature and precipitation. There is also a linear growing trend with time, probably due to the growing popularity of courier service.

▶Is it possible to predict the number of couriers for next day or for several days ahead based on the historical data?



Exploratory Data Analysis

- Let us have a look at the correlation matrix first. Besides weather characteristics we add also ‘day_of_week_i’ dummy variable with $i = 0..6$. Here ‘day_of_week_0’ corresponds to Monday.
- Largest correlation for ‘courier_partner_online’ is with ‘date’, ‘temperature’, and then with ‘precipitation’.
- Correlation with ‘humidity’ is also large. However, ‘humidity’ is almost deterministically correlated with ‘temperature’. It should not add new prediction power.



Prediction Tasks & Feature Engineering

- **Next-Day Task**

Predicting the value of the y_{i+1} (representing 'courier_partners_online') for the next day *i+1* based on the historical values available up to day *i*.

- **Multiple-Day Task**

Predicting the values of the variable $y_{i+1}, y_{i+2}, \ldots, y_{i+n}$ for the next n days *i+1, … i+n* based on the historical values available up to day *i*.

- **Feature selection**

  - Features consist of historical variables '*courier_partner_online*' '*temperature*', '*precipitation*', '*day_of_week_0*', … '*day_of_week_6*'.
  - Look-back window size: '*train_days*' parameter controls number of preceding days to be used as features. We take typically 40 days to describe both small- and meso-time scales like season variability.
  - With 10 feature and 40 preceding days, the total number of features is 400.

- **Additional features**

In the notebooks you can experiment adding more features such as '*date*', periodic '*day_of_year*', '*day_of_week*'.

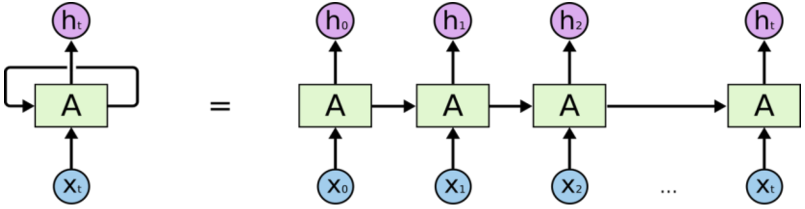
Models

Baseline - Linear Regression (LR) Model:

- **Parameters**: Equal to the number of features (400)

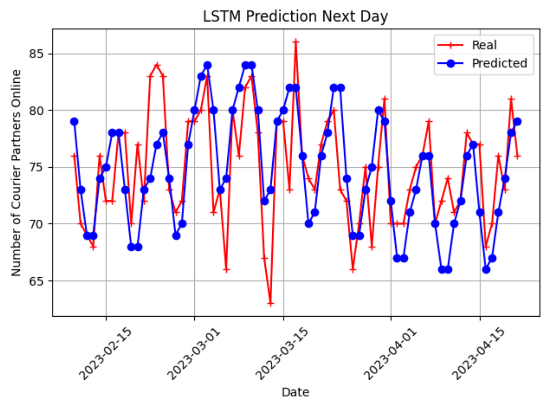
Long Short-Term Memory (LSTM) Model:

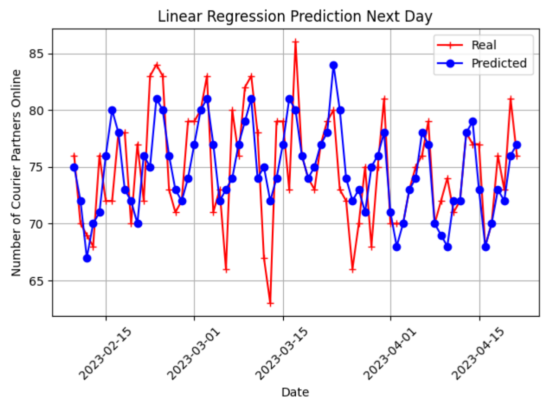
- **Architecture**:
  - **Input**: (timesteps, features)
  - **Layers**: 4 LSTM layers with varying units and dropout (30%, 10%, 20%, 30%)
  - **Dense Layer**: Output layer with n units, which can be 1 or e.g. 20
  - **Total params:** 138,271
  - **Loss**: MSE, **Optimizer**: Adam

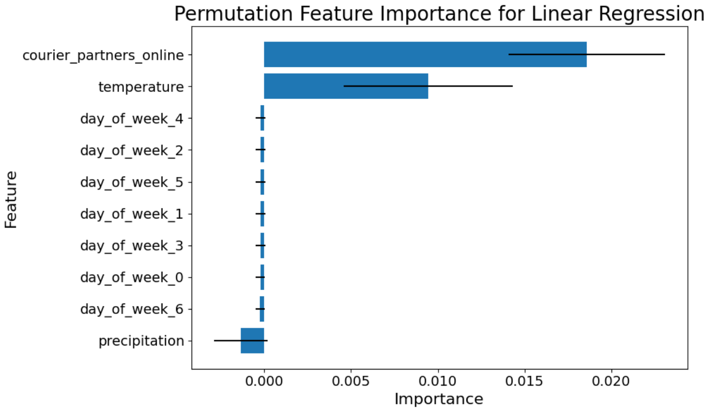


We adopt the [LSTM realization for stock prise prediction]()

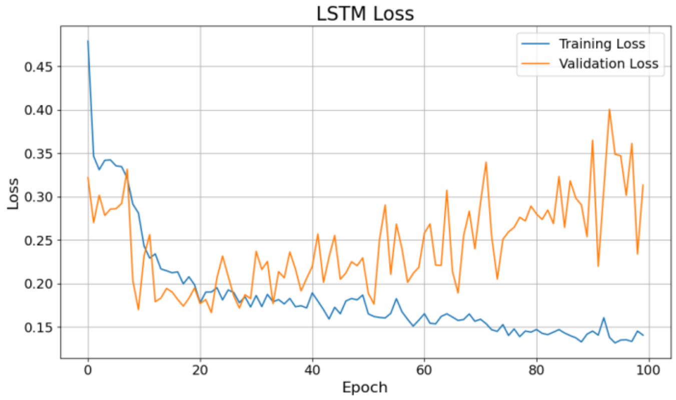
Results: Next-Day prediction







Learning curves (see in /notebooks)



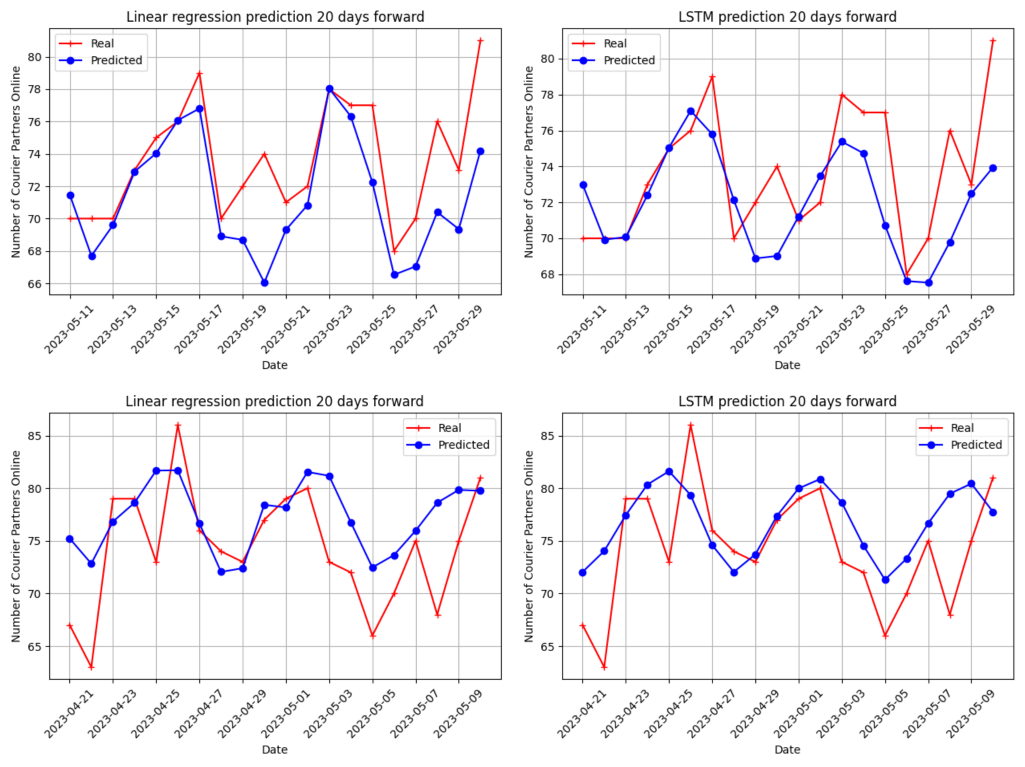
Residual errors

Model	MAE	RMSE	SNR	R2
LSTM	2.92	3.93	25.61	0.35
LR	3.04	3.90	25.67	0.36

- Linear Regression works a bit better than LSTM
- Most important features: ‘courier_partner_online’, ‘temperature’
- Optimal training stop is at 20 Epoch

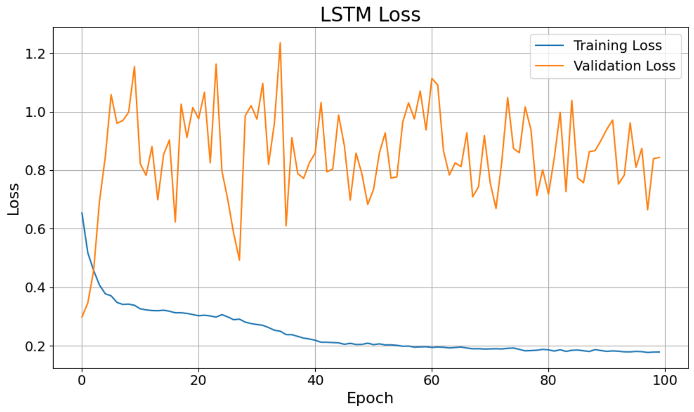
Results: Multiple-Day prediction for 20 ahead

Predictions for different 20-days terms



residual errors

Model	MAE	RMSE	SNR	R2
LSTM	3.43	4.32	24.95	0.29
LR	3.51	4.36	24.71	0.17



- LSTM performs better than LR
- However, LSTM overfits so that test and validation errors fluctuate. Metrics can vary from run to run

Conclusion

- Both LSTM and Linear Regression models predict courier numbers for the Next Day and for Multiple Days ahead based on the historical features.
- For both tasks metrics RMSE, SNR, and R-squared show decent performance (R-squared ~0.2-0.3). For the Next-Day task, both models perform similarly, while Linear Regression slightly outperforms for the Multiple-Day task.
- LSTM performance depends on hyperparameters like batch size and early stopping. For Multiple-Day task LSTM model overfits. Future improvements can focus on hyperparameter optimization and expanding the time series dataset to further enhance performance.