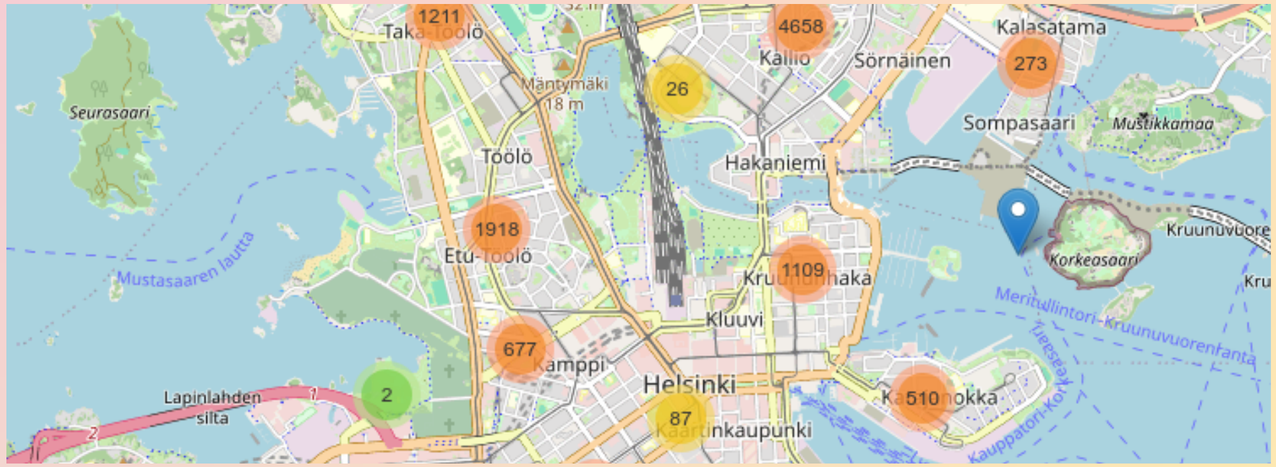
Prediction of Customers in Different Parts of Helsinki using Deep Learning

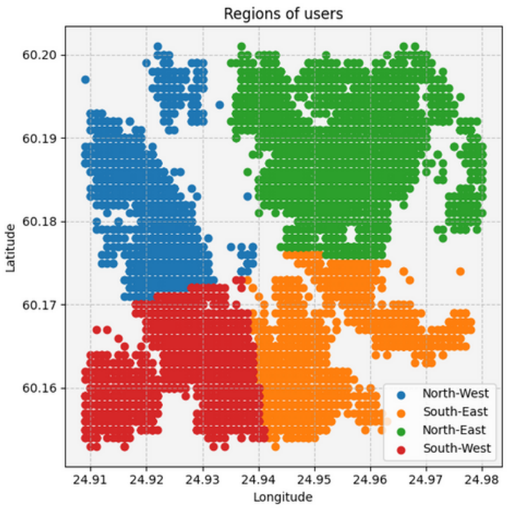


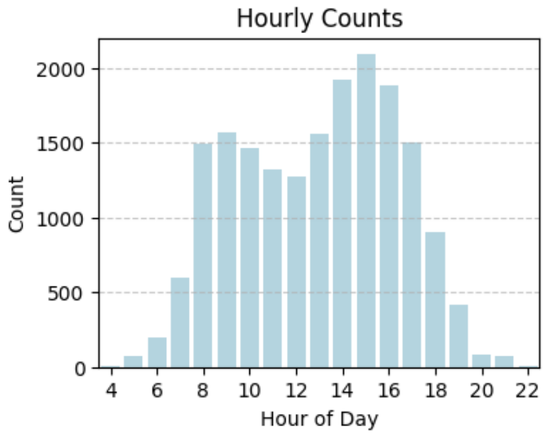
DANIEL WOHLRATH

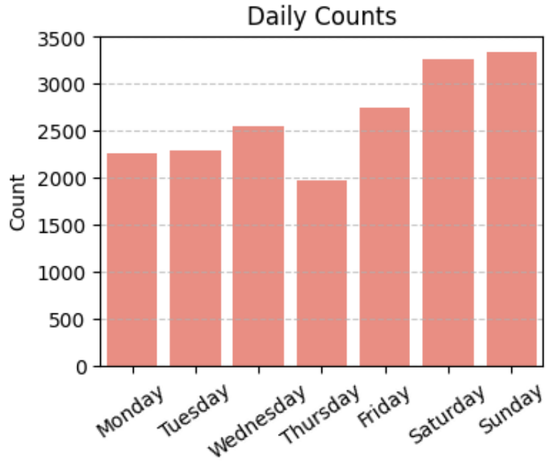
JANUARY 2024

Main Objective

Where will people order from in the next hours?







Understand needs of customers

Where and when should couriers go online?

Make predictions

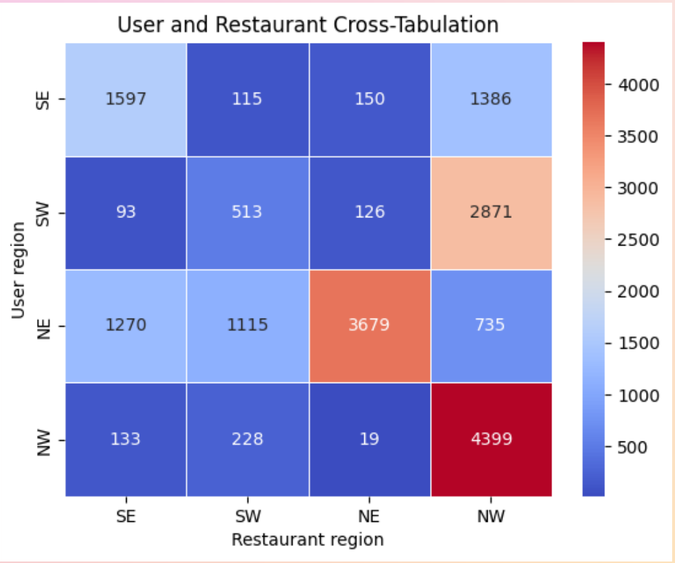
based on weather, day of the week, hour of the day

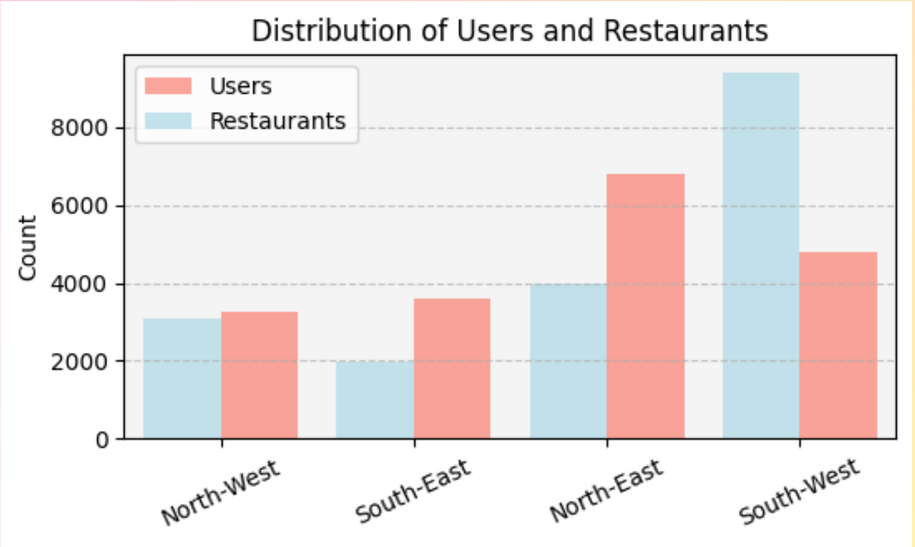
Partnerships

Close partnerships in areas of high demand?

Locations’ Insights Utilization

- Users in S-W order 6x more from somewhere else
  - **This is an opportunity for a new restaurant partnership in S-W!**
- Users from S-E order a lot from their own area, even though there is not many restaurants.
  - **This suggests having more couriers in this area.**
- Whether it is raining, the weekend, or 9 in the afternoon, stay one step ahead.
  - Predict orders with following models:





Data Preparation

Features: Day, location, cloud coverage, temperature, wind speed, precipitation, hour

Target: Number of orders

One-Hot Encoding: Location is one-hot encoded

Scaling: Rest scaled into range [0,1] using MinMax for increased performance

Data Split:

- Test: last 7 days,
- Validation: preceding 5 days,
- Test: rest

Loss: MSE in Model Selection for (RF, K-NN, XGB) and also for total evaluation of goodness together with MAE

Used Models (Pt1)

Multilayer Perceptron

- 8 hidden layers with robust ReLU nonlinearities.
- Inclusion of 20% dropout between every other layer as a safeguard against overfitting.

Bidirectional LS-TM

- Long Short-Term Memory Recurrent NN can uncover intricate temporal patterns in time series prediction tasks.
- Sequence of 8 layers of Bidirectional LS-TM,
- more nuanced understanding of complex patterns.

Used Models (Pt2)

Random Forest

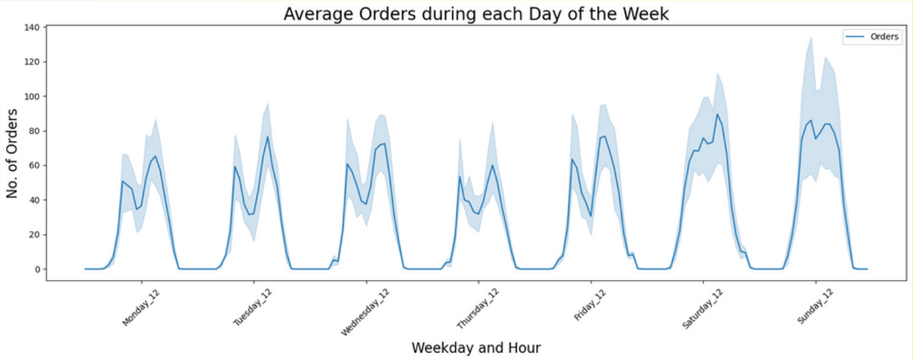
- combines multiple decision trees to improve predictive accuracy and control over-fitting.
- Hyperparameter tuning (HPT) using GridSearch
- number of trees: 5000
- maximal depth of a tree: 10

XGBoost

- Flexible Gradient Boosting Algorithm
- well-suited for time series data due to its ability to capture complex trends over time.
- HPT using Grid Search
- learning rate: 0.1
- number of boosting stages: 300

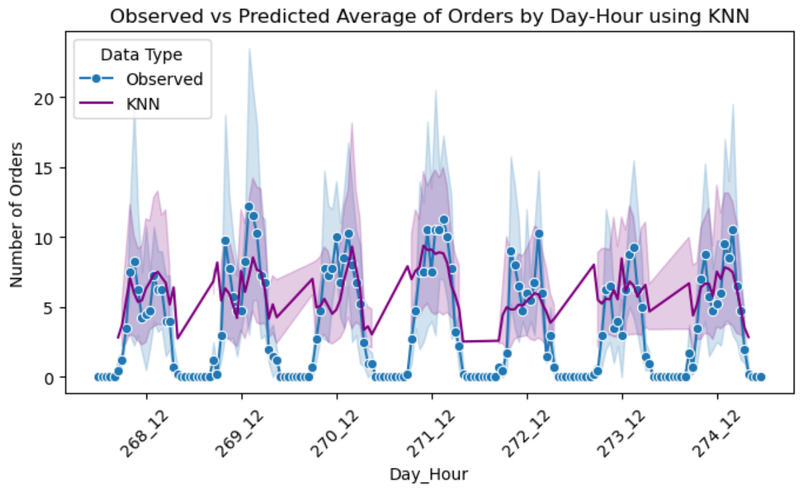
Expected Behaviour of Prediction Models

- Prediction model should follow bimodal shape
- Be able to predict more orders during weekends



K-Nearest Neighbours

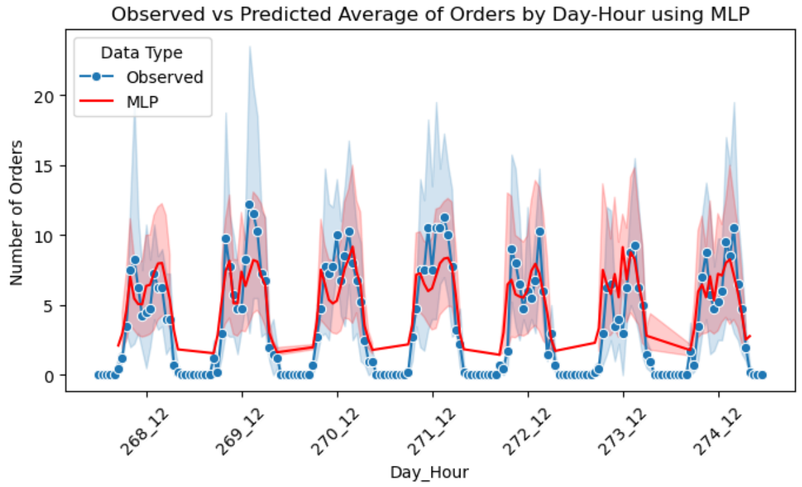
- highly effective in capturing local patterns within the data.
- HPT using GridSearch
- number of neighbours: 7

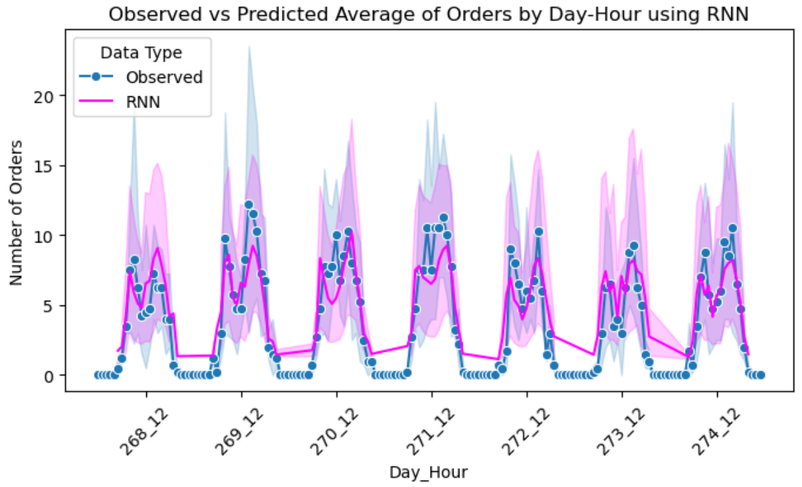


Average number of orders using bootstrapping

The line graphs illustrate the trend of average number of orders over time.

The filled area around is a 95% confidence interval created using bootstrapping.



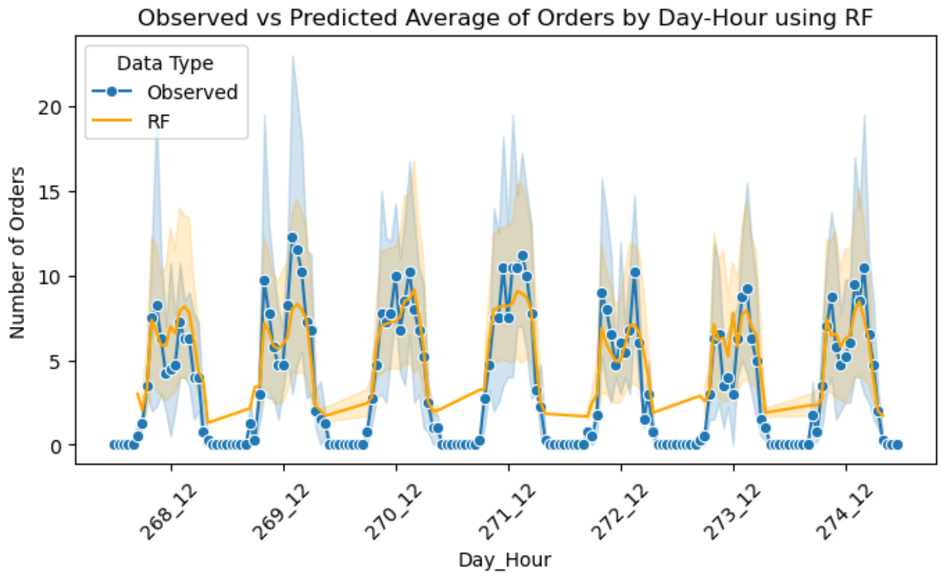


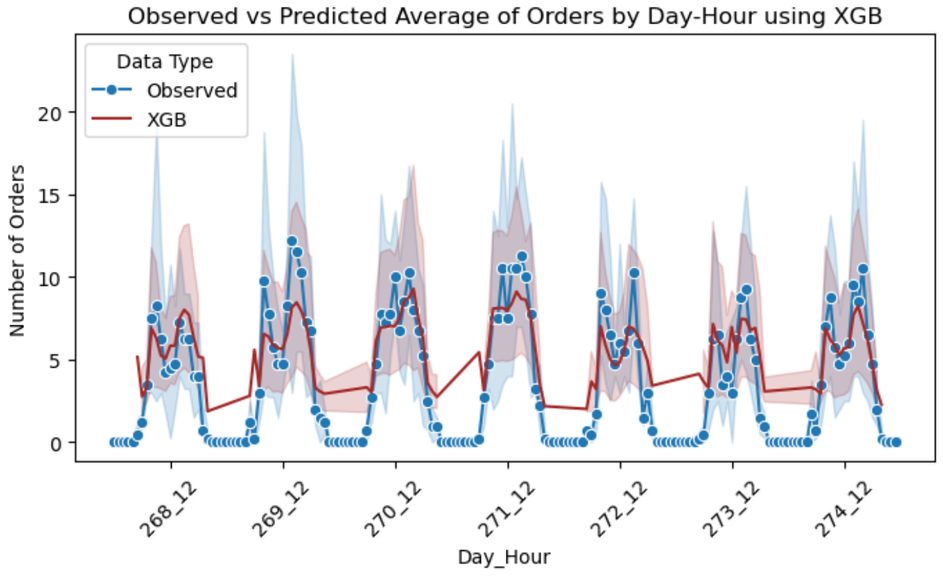
K-NN, MLP and LSTM

The test data is accuratelly replicated on a satisfactory level.

The observed data is padded during the day for visualizations purposes.

Good fit is desired only during the days, not during the nights.





Average number of orders using bootstrapping

The line graphs illustrate the trend of average number of orders over time.

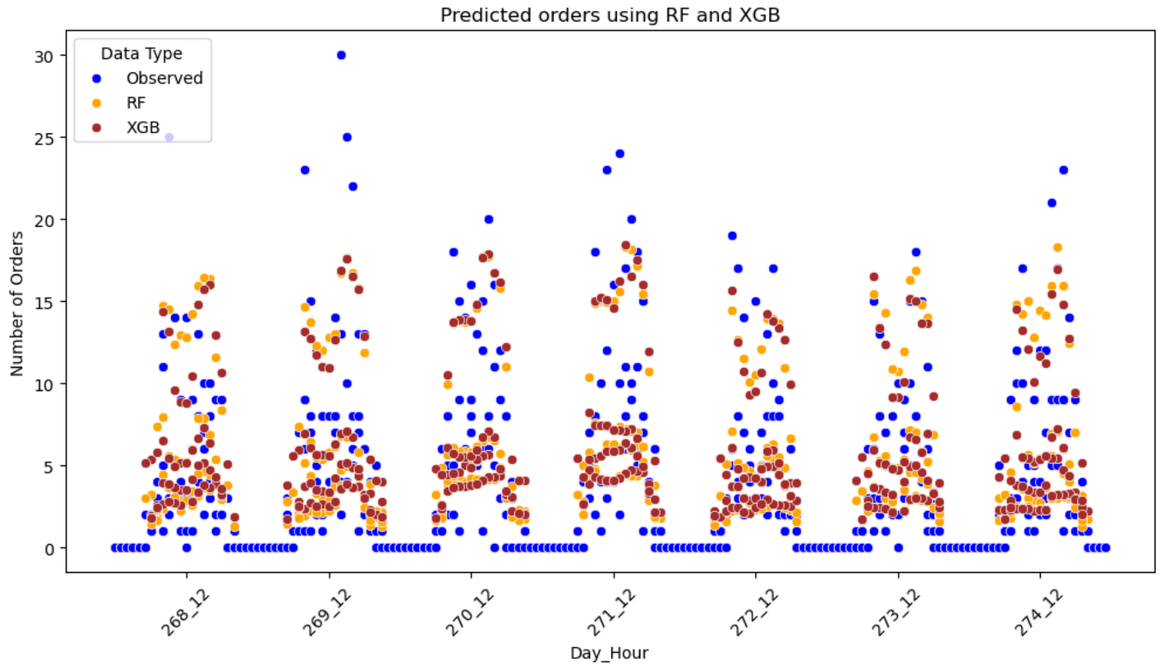
The filled area around is a 95% confidence interval created using bootstrapping.

RF and XGB

Random Forest and XGBoost exhibit excellently predicted patterns on unseen data.

The bootstrapped mean confidence intervals are almost identical to those observed

The fit during night hours is irrelevant, as test data has been padded for visualization purposes



Test Loss	Random Forest	XGBoost	LS-TM	MLP	KNN
MSE	7.12	7.74	9.95	10.76	11.06
MAE	1.96	2.08	2.33	2.37	2.57

Evaluation

Random Forest is able to accurately learn underlying patterns from data and predict with the highest accuracy on test data. It is also reliably learning the bimodal type of the desired orders' distribution.

XGBoost is the second best performing algorithm. The boosting nature of the model is attractive concept for obtaining high accuracy models and is very flexible.

The plot shows observed and predicted number of orders using Random Forest and XGBoost.