HSL Suomenlinna ferry traffic predictor

Gabriela Kakol, Christie Netto, Jing Cheng

Introduction

Suomenlinna, a UNESCO World Heritage site, is one of the most popular destinations to visit for both tourists and locals. You can enjoy the island for its rich history, cultural attractions, or simply a scenic picnic spot. Tens of thousands of people visit the island every year, and around 800 people residents on the island permanently[i]. It is located near the capital of Finland, Helsinki via a short ferry ride, the only transportation option for the general public as well. The ferry is organized by HSL (Helsinki Regional Transport Authority). It runs from Kauppatori (Helsinki Market Sqaure) to Suomenlinna from 6am to 2am all throughout the year. The journey is only a short 15-minute ride.

Due to its popularity and limited public transport options, planning the travel ahead is essential for both tourists and locals. Therefore, we offer the HSL Suomenlinna ferry traffic predictor, a way to manage your trip efficiently and accurately according to the traffic busyness level.

This report outlines the technical aspects of the application, including data collection and preprocessing, modelling, app design, visualization, conclusion, and potential future improvements.

Motivation and added value

The HSL ferry ride between Suomenlinna and Helsinki, given the fact that it is one of the only transportation options for public, the importance of an accurate predictor is undoubtful for both tourists and inhabitants. The traffic can be affected by various factors, for example the weather or specific hour during the day. Therefore, we aim to create an easy-to-use, accessible tool for people that want to travel to Suomenlinna island in an optimal manner traffic wise.

The motivation of this project - HSL Suomenlinna ferry traffic predictor, is to predict ferry traffic between Helsinki Kauppatori and Suomenlinna through time and weather conditions. The goal is reached by training a model for predicting passenger amount, and then visualizing the traffic predictions in chart for end-users. This can assist the traveler to plan their schedules, optimize routes, and improve overall passenger experiences.

There are various benefits from using the prediction model, which add value to everyday life. It includes operational efficiency for the HSL, as the app can help them optimize the ferry schedule, reduce crowds and further allocate resources efferently during busy times. It not only offers an insight for HSL operators, but also to the Helsinki city tourism management. They can tailor their services accordingly and increase or decrease the frequency of tourist services during certain time periods. As previously mentioned, it can enhance end-user convenience, so passengers can avoid overcrowding on the ferries and plan an optimal journey.

Data collection and privacy

Two datasets are used to build the predictor model:

Suomenlinna ferry passenger data from HSL

- REST API dataset: https://hsl.louhin.com/lati/help [ii]

Weather observation data from Finnish Meteorological Institute

- CSV dataset: https://en.ilmatieteenlaitos.fi/download-observations [iii]

Having downloaded the dataset from open source, it was easy to then modify it for the next step.

Ethical consideration and privacy protection are important in the conducting and designing of our web app. All data are extracted from open public datasets. It does not contain private data indicating the identity and information of individuals. We collected the data and conducted the design of our predictor under the GDPR (EU General Data Protection Regulation) guidelines. And we do not use it for commercial purposes. Therefore, pseudonymization and anonymization of the data can be assured.

Data preprocessing

Data preprocessing is an important and crucial step in the design of our application. It transforms the raw data into a predictable and analyzable form.

The HSL data file was downloaded as a csv file. It contains a lot of variables, which we filtered for the ones we need: Year, Month, Day, Hour, Stop Direction (to/from Suomenlinna) and number of Passengers. We also translated the open dataset from Finnish (‘VUOSI', 'KUUKAUSI', 'KUUKAUSIPÄIVÄ', 'TUNTI', 'PYSÄKKI', 'NOUSIJAT') to English as our service and app is offered in English.

For the weather observation data, we downloaded it from Finnish Meteorological Institute in a csv file. It contains the Year, Month, Day, Time, Average temperature (°C), Wind speed (m/s), Precipitation (mm). In which we take the Average temperature (°C), Wind speed (m/s), Precipitation (mm) variables, and processed the Time (xx:xx) to split the hour and minute, then only take the hour.

We first clean the data by essentially removing the unnecessary variables and choosing the ones that are needed for analysis. We then combine the two processed data files. The variables that exist in both datasets include the Year, Month, Day, Hour will be used for merging and aligning of data. Both datasets are time-series data, and this step ensures that only rows corresponding to the same date and hour will be merged. The merged dataframe contains information from both HSL and the

weather observatory at the same hour and date. We then convert the 'Average temperature', 'Wind speed', 'Precipitation' into numeric float value. It any value cannot be converted; we replace it with 0.
Disclaimer: we will only use the whole-year ferry route data that operates between Kauppatori and Suomenlinna.

Learning task and approach

Our learning model is based on regression. *Regression analysis is used to model and estimate the relationship between a dependent variable (*in our case, the number of passengers) *and independent variables (*weather and time factors) *statically* (Sebastian Taylor, n.d.). This is an ideal method to reach our goal of predicting the number of passengers on the ferry during a specific time and on specific route, given the historical weather and traffic input factors.

The models we used include the Standard Scaler and Random Forest Regressor model. Where Standard Scaler is used to standardize and ensure that all the data are contributed equally to the model, without the case of one large scale data dominating others. This is important as a lot of variables are presented in different scale, for example, temperature is measured in degrees (°C), and wind speed in meter per second (m/s). Without scaling, one variable could dominate the learning process and lead to inaccurate predictions and outcomes. Random Forest Regressor was chosen as it is an ensemble learning method that is capable of handling complex datasets, it essentially builds multiple decision trees during training and aggregates their results to make predictions. It improves accuracy and reduces overfitting. In practice, the number of passengers might increase non-linearly with the temperature variable and Random Forest Regressor can help capture these non-linear patterns and handle it accordingly.

The training is conducted by splitting the dataset into the training set and the test set in an 80:20 ratio respectively. This means that 80% of the dataset is used for training the model, whilst the remaining 20% is used for testing the performance. In this process, we use a machine learning technique, cross validation, to hyperparameter tune the Random Forest Regressor model. Hyperparameter essentially controls the learning process, functions and end result, playing a crucial role in the fitting of the model. The feature importance analysis was also utilized to identify the level of significance that each variable has on the overall ferry traffic output. Some variables include, for example, the temperature, participation or the time.

Visualization and Communication of results

We want users to be able to easily make predictions and access the results. Hence, we created a desktop application, which can be easily downloaded and run. The user must download the folder with the files and install the application by running the following commands:

poetry install
poetry run invoke build

Applications can be easily initiated by running command: *poetry run invoke start*.

The application has 2 windows: main window and results window. In the main window, the user is asked to specify details regarding their journey on the Suomenlinna HSL ferry: the direction, date, time, expected weather, wind and precipitation. The user confirms the input data by clicking the button "Predict the traffic" on the bottom of the window. Clicking the button activates the prediction model in the backend, which runs with the input data specified by the users. It also redirects the user to the next window – the results window, where the predicted ferry traffic is displayed. The scale of the busyness is as follows:

- Not busy: up to 80 passengers
- Moderately busy: 81 – 160 passengers
- Quite busy: 161 – 240 passengers
- Very busy: 241 – 320 passengers
- Extremely busy: over 321 passengers

The displayed scale is also colour coded, and the corresponding colours are: green, yellow, orange, red, burgundy. Below the result scale, the user can see the parameters for which the prediction was made. Moreover, by clicking the "Make another prediction" button, user can go back to the main window and start the process again to, for example, predict the traffic for another day.

The front-end and UI were developed using Tk library in Python. The code is located in the Suomenlinna-Ferry-Predictor/src directory.

Conclusion and Improvement

In conclusion, our project was conducted successfully as we were able to implement the initial core idea of finding an optimal time for travelling to Suomenlinna by predicting the busyness of ferry traffic. The final model can offer accurate predictions based on weather and time variables input. However, throughout this project we did encounter issues as well. For example, in deciding and finalizing the visualization aspect. We also faced challenges in the data processing, particularly in the merging of datasets, and scaling.

While we achieved our established goal, improvements could still be made in the future. The current model does not differentiate between public holidays, weekdays or weekends, which are believed to post an effect on ferry traffic. We can also enable predictions for more than 1 year ahead, as the current model can only predict till the year of 2025. Another area we would like to optimize is the speed of the process. Although the Random Forest model is easy to train and is suitable for our project, the large number of decision trees can slow down the operation and affect

efficiency negatively for real-time predictions. In the future, we would also want to expand the app to a mobile version, so that it is more accessible for end users.

References

i: Suomenlinna Sveaborg. (n.d.). *People of Suomenlinna*. Suomenlinna.fi. Available at: https://www.suomenlinna.fi/en/fortress/people-of-suomenlinna/

ii: Helsinki Region Transport (HSL). (n.d.). LATI Help. Available at: https://louhin.hsl.fi/lati/help

iii: Finnish Meteorological Institute. (n.d.). *Download observations.* Available at: https://en.ilmatieteenlaitos.fi/download-observations

Taylor, S. (n.d.). *Regression analysis.* Corporate Finance Institute. Available at: https://corporatefinanceinstitute.com/resources/data-science/regression-analysis/

Appendix

Github repository: https://github.com/christienetto/HSL-Suomenlinna-Traffic-Predictor