Machine Learning Primer 21.10.2024
WiSe 24/25 **C. Gros**, D. Nevermann

Project #1

Train-Pain at Frankfurt Central Station?

Deadline: 11.11.2024, 12:00h

What are the semester projects? The semester projects are supposed to challenge you with medium-sized machine learning problems that would be too excessive for a weekly exercise sheet. The projects will mostly be not as well-defined as a regular exercise, leaving you room to go your own direction.

Rules

- Semester projects need to be handed in by the assigned deadline. You have 3 weeks for a project.
- Hand your projects in via mail to your tutor with all relevant files enclosed.
- You can work in groups of up to 3 people.
- You have to pass at least 3 out of 4 projects.
- When asked, you have to be able to present your project in front of the tutorial group and answer questions. Otherwise, the project does not count as passed.

What is the project about?

Did you ever wait for a delayed train at Frankfurt central station? Did you ever wonder if it is just you that always has bad luck and picks delayed trains? Let's find out![1]
In this project you will analyze data of train departures from Frankfurt central station. The goal is that you get familiar with the Python programming language and it's numerics and plotting libraries.

Where do I get the data from?

A big part in training supervised learning models[2] for machine learning tasks is obtaining data. Although in this project you are not training machine learning models (yet), it is your task to obtain data.

Downloading train data Data about departures of Frankfurt central station is most easily obtained through an API[3]. The Deutsche Bahn provides an API to access train departures, arrivals and much more, see here. From my research, the easiest way to interact with the API in Python is through the pyHaFAS package using the DB profile, see here. You are, however, free to use any way you want.

What should I download? The goal of the project is to analyze departures from Frankfurt central station. Download data for at least one weekday of departures of long distance (e.g. ICE, IC), regional (e.g. RE, RB) and suburban (S-Bahn) trains. If you are interested in the project, you can of course go further (e.g. compare a weekday and a Sunday).

Important Warning!!! APIs have terms of service that regulate acceptable interaction. It is your responsibility not to break any rules.

Do not spam requests to the API!

Downloading data every few seconds or even more often might not only cause problems, but is also completely unnecessary (trains don't leave in a matter of seconds, we are still talking about the Deutsche Bahn here...). Choose a

[1] This project is inspired by the highly recommended talk *BahnMining* by David Kriesel, see here.

[2] Supervised learning means that the model is trained with input-output data pairs.

[3] API stands for **A**pplication **P**rogramming **I**nterface. APIs are interfaces provided by companies that let users easily interact with a service through code.

wise download frequency. And just to be extra sure, use your own network connection. Then it's definitely not our problem ☺.

What should I analyze?

Using your data, answer the following questions:

- What are the rush hours of the day, where the most trains depart?
- When in the day are the most trains canceled?
- What does the delay distribution look like, and how do delay times and cancelations of different train types (intercity trains, regional traffic and suburban trains) compare?
- How is delay related to the train density?
- What is the probability that trains after a delayed train are also delayed?

You are of course free to get creative and think of interesting questions yourself.

Minimum requirements

Your project should fulfill some minimum requirements:

- Use Python for your project.
- Download at least one workday of data and answer the questions above.
- Don't get yourself arrested by accidentally ddossing the DB ☺.
- Present the results of your analysis in a suitable way (written report, slides, Jupyter notebook, ...).

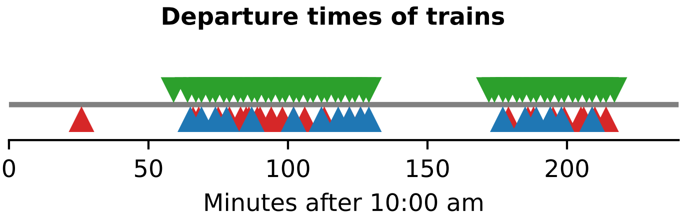
Hints and example

Some hints For plotting I suggest using Matplotlib. For managing the downloaded data, Pandas can be helpful. An all around useful library is NumPy. If you are already aiming for ML related ways to work with your data, use PyTorch or Scikit-learn.

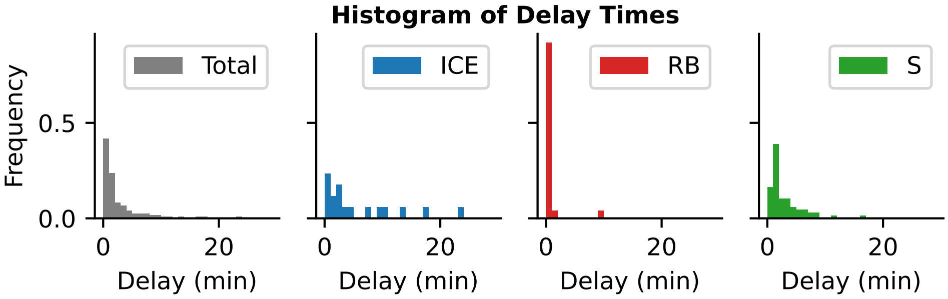
Example You can find a small example dataset here. The data is stored as a CSV file covering roughly 3 hours of train departures. Columns represent (in order)

train ID, train name (e.g. ICE123), destination,
departure time, if canceled, delay, platform.

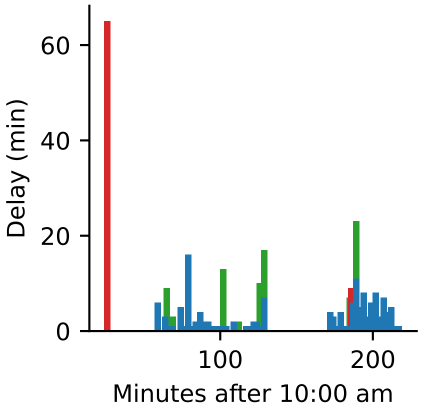
The data contains duplicate trains. Below are some example plots generated from that data.
The data was downloaded in multiple batches. This is the departure times after 10:00 am, where different train types are color coded (see below):



Here is a histogram of delays for the different train types:



And finally, a plot of the delays as a function of the departure time:


Delay (min)
Minutes after 10:00 am