Assignment 3

Ludvig Flodin ludflo-0

April 2025

Data preparation

The data pre-processing was:

- Labeling
- Remove missing values
- Getting coordinates for departure/arrival
- Havestine distance between cities
- Season
- Holiday

This task is about how we can model the estimation of a delay of trains. The dataset contains all the reported delays between 2018 and late 2020, where a a delay is considered 3 minutes late. I based my labeling of SJ's own ticket RMA and my own gut feeling.

- Less than 5 minutes is labeled punctual
- 5-20 minutes is labeled late
- 20-40 minutes is labeled very late (50% cashback of ticket)
- Above 40 mintues is labeled extremely late (75% cashback)

Removing missing values was made using pandas function 'dropna' which reduced the dataset from 17834 to 16699 rows, where 'Reason code 3' still contains missing values. After removing the missing values from 'Reason code 3', the data set was only 7237 rows. So I decide to keep a separate dataset containing the missing values.

To get the city coordinates, I used a plugin called GeoPy which gives the city coordinates from a single string. This was made with a loop based on the route column from the train dataset. I manually did a lookup table that translates the city akronym.

Instead of using pythagorean theorem, I used havestine distance which is a more exact way (or overcomplicated) way to get a rough calculation of the disctance. Havestine distance takes the earth curvature into consideration, so maybe not necessary for relatively short distances.

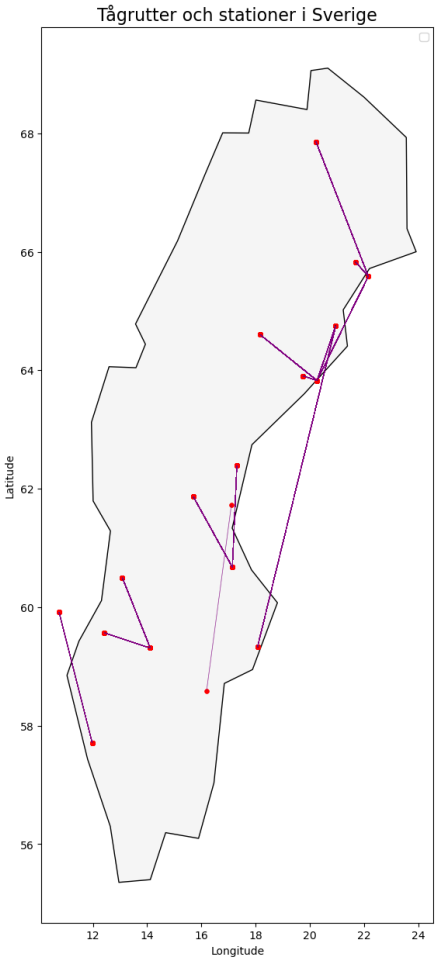
Season was added because I do not manage to import consistent data from SMHI. This season column was soley based on the time of year.

I thought that holiday may have some importance here, because working force may be limited. I added a column with the help of the library holidays.

Feature Engineering / Visualization

These are the parameters I assumed to have an effect on delay.

- Route number
    - If a specific route is more troublesome
- Departure City
- Departure Coordinates
- Arrival City
- Arrival Coordinates
    - For visualization?
- Tågnr
    - If a specific train cause more trouble
- Reason code Level 2
- Reaason code Level 3
    - If a specific reason may contribute to severity of delay
- Season
    - If one season contributes to more delays than another
- Havestine distance
    - If distance of route contributes to delays
- Holiday (Bool)
    - Workforce may be weakened on Swedish holiday dates



Reason code level 3

I began to look at how the error code affects the delay. These are the top 15 contributing reason codes. (Figure 1), but the sample rate of Verkstad överskrider tid was as low as 2. So for Figure 2, I filtered the dataset so that only categories over five data points (rows) was shown. Besides wheel damage and faulty brakes, a lot of the delays comes from administrative/logistic delays. And stated in the class, wheel damage is caused by an imperfection of the track, resulting in a faulty wheel that breaks over time.

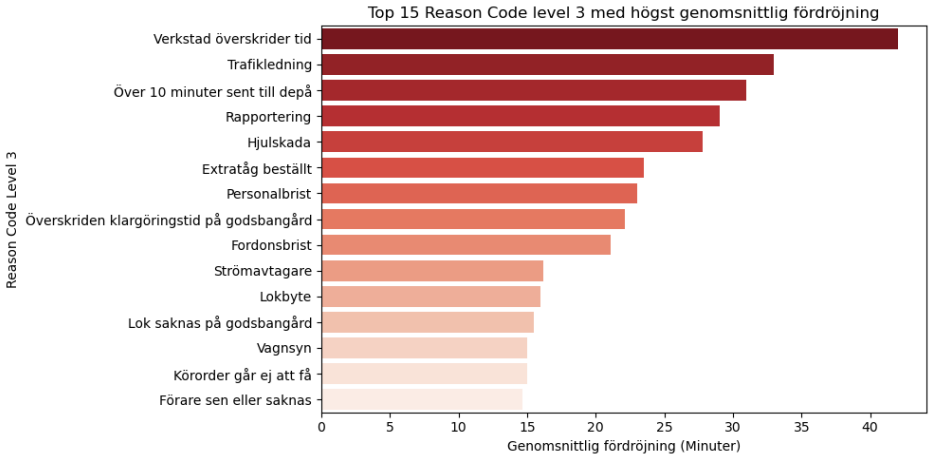


Figure 1- Top 15 Reason code 3 to mean delay

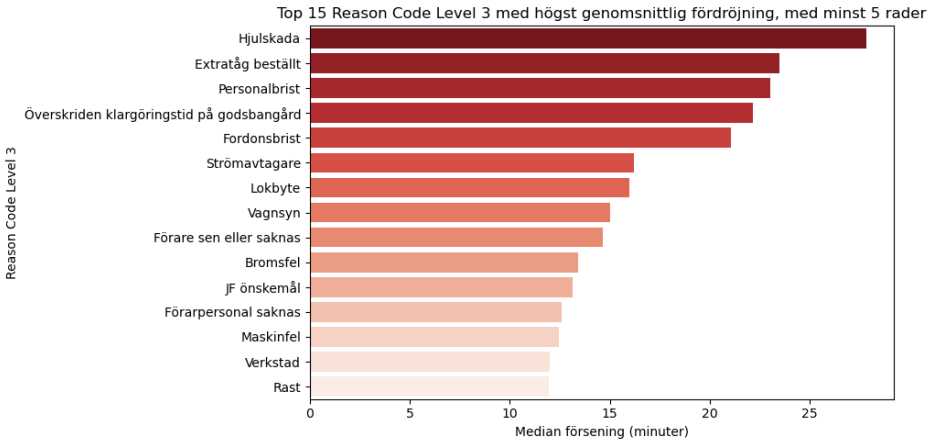


Figure 2 - Top 15 reason code 3 to mean delay (with atleast 5 rows)

Havestine distance

The distance of the route does not seem to correlate to delay time as Figure 3 shows. When looking closer to mean delay vs route (figure 2) we can see that the mean delay spans between a few minutes. I think that this is also backed up by the fact that most of the delays comes from an administrative or logistical standpoint.

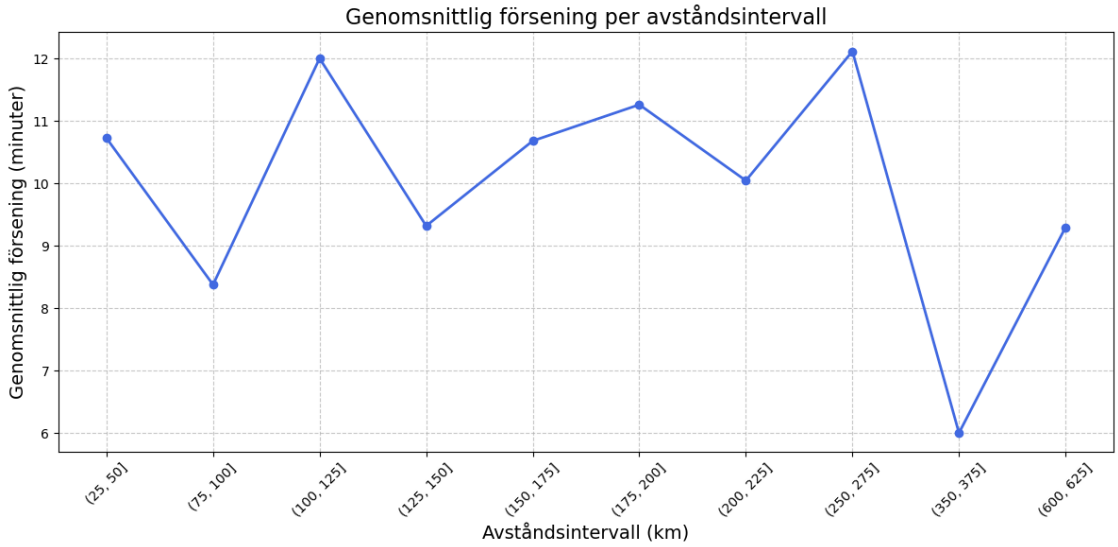


Figure 3 - Mean delay vs Distance intervals

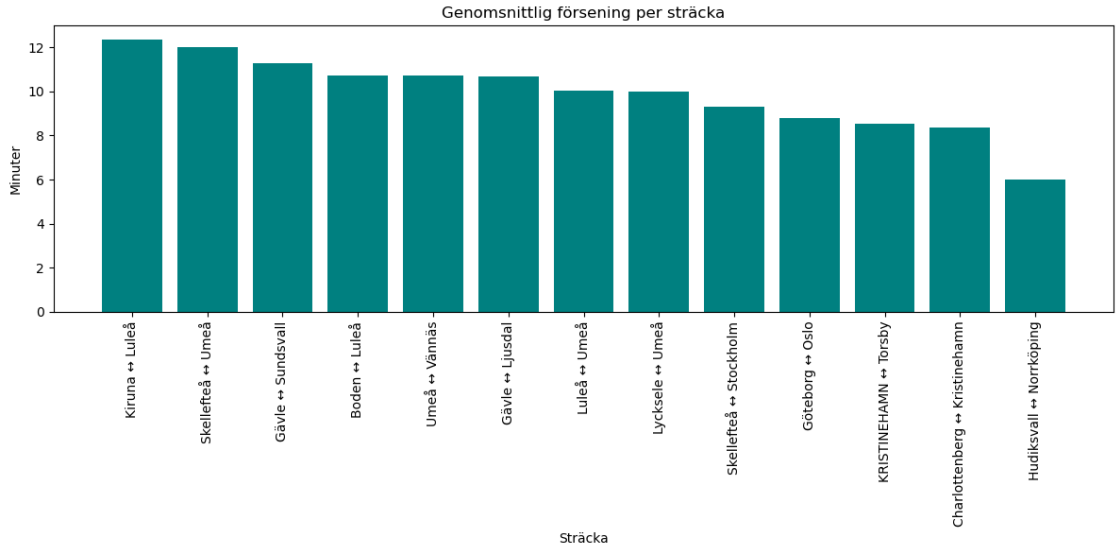


Figure 4 - Mean delay by route

Seasonal or holiday factors

I thought that season might come to play if there is a delay or not. As figure 5 shows, the dataset is a bit scewed towards winter. This might be because of my subjective definition of what months winter is, as in my opinion it spans four months.
When it comes to holidays, which dictates about 5% of the total number of days of the year, we can see in figure 6 that delays may be more frequent during holidays.

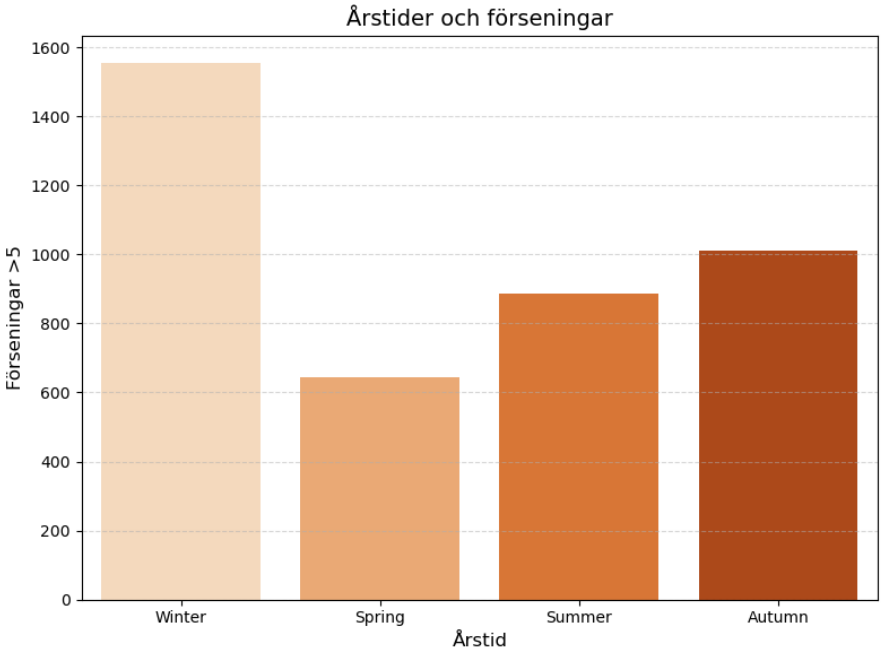


Figure 5 - Delays per season

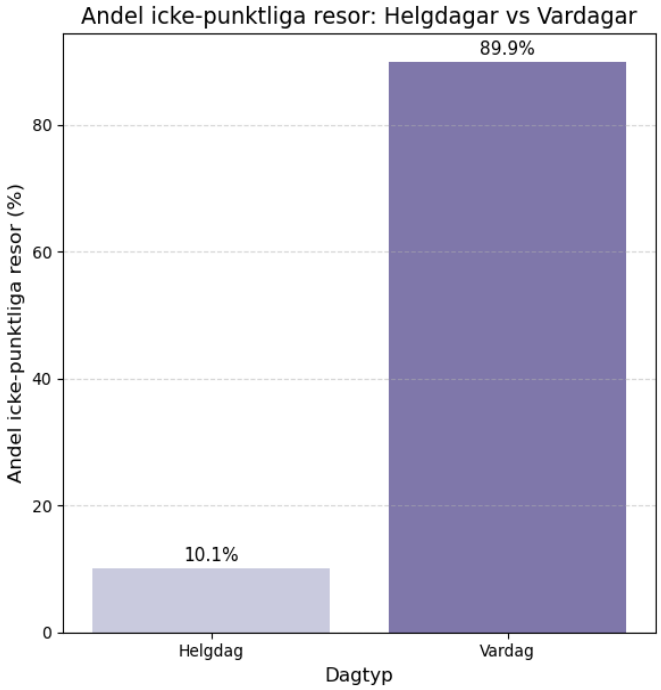


Figure 6 - Delays on holidays vs normal days

Random Forest

For the first classification I tried random forest. I realized that I had to do more pre-processing because random forest cannot handle strings as features. So what I did was to encode every string into new columns that contained Booleans. For the labeling, punctuality, I used sklearn label encoder, which translates the different strings intro its own integer.

When testing with stock settings of the random forest, it achieved a shy accuracy of about 40%, but when removing all missing values from Reason Code level 3 it achieved an accuracy of about 60%. A reason