**Experimentation**

**Overview**

To test the data, a wide range of models are used. In each, their usage is compared to uses previously identified in the literature. The concept behind the model is explained and the method used is detailed. Results are then thoroughly discussed. Both classification – whether or not a train is delayed – and regression – by *how much* a train is delayed – are addressed. Finally, the influence of weather will be tested.

**Pre-processing**

Pre-processing is necessary before using the data output from the LOAD stage. This mostly takes the form of encoding categorical variables, dropping unnecessary columns, and setting datatypes to save space.

Special attention is paid to datetimes, as they are likely to have significant influence. Dates are split into their respective parts: year, month, day, day of week. Hour and minutes are encoded into a single minutes variable. Different variables are used for both origin and destination. These are encoded cyclically[[1]](#footnote-1) by converting a single column to two: one for and the other for . In this way a variable is encoded as a unit circle, so Monday and Sunday are adjacent, for instance, as are 11pm (23 hours) and 12am (0 hours).

Two target columns are then constructed using the difference between the scheduled arrival time (STA) and actual arrival time (ATA). The first, whether or not a train is delayed, is used for classification. The second, the amount by which a train is delayed, is used for regression. Erroneous values are also discarded. In about 1% of records, a error in the ETL pipeline resulted in ATAs a day *before* STAs. This occurred on a small number of trains that run over two days.

Some element of locality is also likely to improve results. Categorically encoding origin and destination TIPLOCs is impractical, as there are approximately 80,000. Nor is there any structure to their designations. Instead, their corresponding STANOX area is used. There are 89 STANOX areas[[2]](#footnote-2), proxies for geographical areas.

Dropped columns are mostly IDs, and those used to construct the columns discussed above. The actual time of departure (ATD) must also be dropped, as it could not be known in real-life data ahead of time. Also dropped are three metadata columns with too many NaN values: *sleepers*, *reservations*, and *branding*.

Finally, the following column are one-hot encoded: *status, category, power\_type, timing\_load, ATOC\_code, seating, origin\_stanox\_area,* and *destination­\_stanox\_area*. These were briefly discussed earlier. The resulting dataset comprises y rows and x columns.

**Regression**

Initial results (using only two days’ worth of data) were very promising: better than 90% accuracy on out-of-the-box sklearn regression models. This is misleading, however. Approximately 7% of trains are delayed. The models have simply learned to predict ‘not delayed’, an example of the accuracy paradox.

That said, decision trees often perform well on unbalanced datasets, and did so here.

Accuracy is a misleading metric for imbalanced data sets.

Classifiers are evaluated using several metrics.

Better to use precision and recall.

Recall is the *true positive rate*.

*Precision* is the ability of the classifier not to label as positive a sample that is negative.

*Recall* is the ability of the classifier to find all positive samples.

Important distinction between metrics that are independent on the prevalence (how often each category occurs in the population), and metrics that *depend* on the prevalence.

The -measure ( and measures) are a weighted harmonic mean of precision and recall. A measure reaches its best value at 1 and its worst score at 0. With , , and the recall and precision are equally important.

For nominal: one-hot, hashing, leave-one-out, target. Avoid one-hot for high cardinality columns and decision-tree based algorithms.

For ordinal: binary, one-hot, leave-one-out, target

Nominal: no order.

Ordinal: order.

Use target / leave-one-out. Preferentially the later. Avoids contamination. Useful for nominal and ordinal data. Classification.

1. https://ianlondon.github.io/blog/encoding-cyclical-features-24hour-time/ [↑](#footnote-ref-1)
2. https://wiki.openraildata.com//index.php?title=STANOX\_Areas [↑](#footnote-ref-2)