Predicting Rail Travel Duration using Machine Learning

Emma Rose Dennis-Knieriem
Instituto Superior Técnico, Lisboa, Portugal
emma.dennis-knieriem@tecnico.ulisboa.pt

May 2025

Abstract

This dissertation contributes to the advancement of predictive modelling in the domain of rail transportation by providing a tailored solution for predicting travel durations on regular services. The proposed model offers a practical tool for improving service reliability and enhancing the overall passenger experience in rail travel. The study begins by collecting extensive datasets from a platform managed by the Federal Office of Transport (FOT) and the System Tasks Customer Information Plus (SKI+) [4] in Switzerland, a comprehensive repository of transportation data, encompassing various parameters affecting rail travel duration. Subsequently, regression models including tree-based methods are employed to develop predictive models. The results demonstrate the effectiveness of the proposed machine learning model in accurately predicting travel durations for passenger trains. Moreover, insights gained from the analysis shed light on the critical factors influencing travel duration variability, thereby offering valuable implications for operational planning and resource allocation within rail transportation systems.

1. Introduction

1.1. Objectives

Accurately predicting a train's arrival time will increase efficiency in rail travel, both for the rail service and for consumer satisfaction. With large datasets, models can be trained to predict train arrival times with high accuracies. One such organization that allows access to its transportation data is the Open Data Platform Mobility Switzerland (ODMCH). Using this publicly available dataset, models were built to more accurately predict arrival times. While this dissertation focuses on rail travel within Switzerland only, the objective is to develop a prediction model that can be used for other countries' rail services, providing that the data collected has the same information used from the Swiss data. With more accurate arrival prediction time, passengers can save time on commuting and do so with more certainty, while rail services can better allocate workers, save on energy costs, and create more efficient schedules, lessening rail congestion and thereby increasing safety.

1.2. Contributions

The Swiss FOT publishes transport data daily. While this dissertation focuses only on trains, the data also include information about bus, ship, and other methods of transportation [1].

This dissertation resulted in the prediction of train arrival times using several different machine learning models. This holds value not only for the Swiss transport system under whose data these models were developed, but for other transportation networks that aspire to adopt more accurate predictions, especially rail systems. These models are flexible in that they can be applied to other data with only basic information: identification of stops, and arrival and departure times.

The Swiss FOT data contains 21 variables, all in Swiss German. For ease of understanding and consistency, they will appear in this dissertation as I translated them into English.

2. Methods

2.1. Data Cleaning

The CSV files for each completed day (January through March, 2021 through 2024) were down-

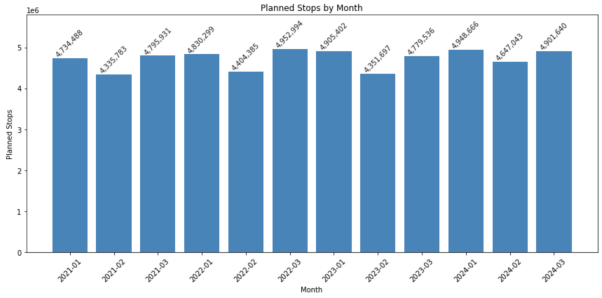

Figure 1: Monthly intended train stops before data cleaning. The dataset had a length of 56,587,864 rows. The number of unique stops was 1,854, and the number of unique lines was 22,967.

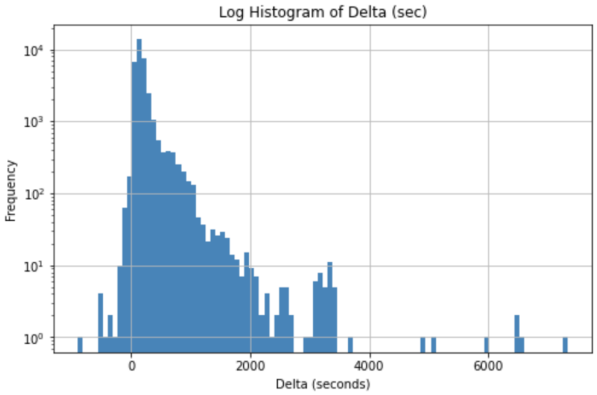

Figure 2: Histogram of Delta times before Delta cleaning step. The frequency is on a log scale for better understanding of data distribution. For counts by Delta threshold, see Table 2.1.

loaded from the open data source. All modes of transportation except train products were filtered out, and the 21 column names were translated from Swiss-German into English. The data were combined into parquet. Once read from parquet files into dataframes, the `OPERATION_DATE` column of type string was duplicated and the copy was converted to datetime format in order to enable grouping and time-based analysis by month and year. The data cleaning steps that followed were removing skipped stops, removing cancelled stops, and removing rows where ETA and/or ETD status was missing. Arrival and departure columns were then duplicated, and the copies were converted from type string to type datetime. Two additional columns were added, indicating the previous stop and its corresponding departure time. This was done by grouping by trip ID and operation date, sorting by arrival time, and shifting the stop name and departure time. After the shift, rows with missing ETA and/or ETD times were removed. A new Delta column of type timedelta was added by subtracting the previous departure time (from the shift function) from the arrival time. A separate column with the Delta values in seconds was also created. Delta values outside the range $120 < \text{Delta} \leq 3600$ were removed (see Table 2.1).

Category	**Count**	**Fraction**
Delta ≤ 0	174,251	0.005
$0 < \text{Delta} \leq 120$	11,306,650	0.328
$120 < \text{Delta} \leq 3600$	22,934,245	0.663
Delta > 3600	6,892	0.004
Delta is NaT	0	0.000

Table 1: Breakdown of row count by Delta threshold (in seconds). Delta values ≤ 0 or ≤ 120 seconds are invalid, likely due to sensor error or a shift function issue. Values > 3600 seconds may reflect rare events such as weather delays. Only rows with $120 < \text{Delta} \leq 3600$ were kept, representing a clean and sufficiently large sample (66% of the dataset).

To prepare for modeling, temporal features were extracted from the departure time into month, day of the week, hour, and minute columns. The stop names and previous stop names were transformed from strings into integers using label encoding [31] for better understanding by the models.

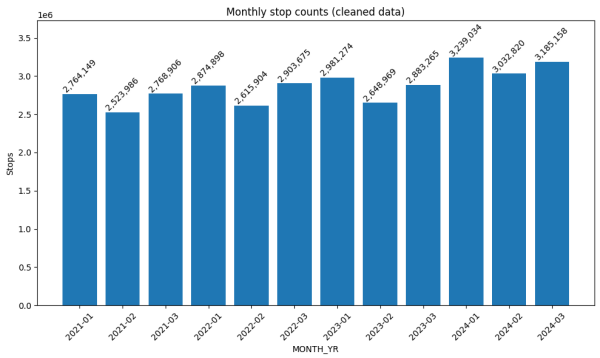

Figure 3: Monthly train stops from cleaned data. The cleaned data length 34,422,038, which is about 60.83% of the length of the uncleaned data. The number of unique stops in the cleaned dataset was 1,480, and the number of unique lines was 19,488.

2.2. Learning Models

Three models were employed: Linear Regressor, Decision Tree Regressor, and XGB Regressor. For each of these models, the training features were the departure day, hour, and minute, as well as label-encoded previous and current stops. The training target was the delta, defined here as time of travel between stations, calculated in seconds.

These three regressors were selected in combination because they each possess different strengths. The linear regressor is the simplest, and can quickly determine if the relationship between features and target is linear. The decision tree regressor is able to capture nonlinear patterns while still being relatively light computationally. XGB can be thought of as an extension of decision trees, since it is an ensemble method that uses gradient boosting. By running simple models alongside more complex ones, this combination gives a wider scope of options from which to select the best model.

A train-test split of size 80% training and 20% testing, with the random seed set at 14 for reproducibility. 5-fold cross-validation [32] in the models was also used, with seed set at 14 for reproducibility. For Decision Tree and XGB, 20 combinations were chosen and run from the hyperparameter space using a randomized search.

2.3. Evaluation Metrics

R^2 and RMSE metrics evaluated the data, and were used to compare to summary statistics generated from the uncleaned data. Model accuracy of the data margins of 1, 3, and 5 minutes were chosen, reflecting acceptable waiting times for passenger transport in Switzerland, where long waits are uncommon.

3. Evaluation

In this section, models' outcomes are displayed in terms of R^2 and RMSE. Feature coefficients are included for the Linear model, while the best performing hyperparameters are discussed for the Decision Tree and XGB models.

For each model, a plot of the actual delay in seconds vs. the predicted delay in seconds is shown, demarcated in color and point shape by thresholds of 1, 3, 5, and more than 5 minutes. Since 5-fold cross-validation was used and only predictions from one fold were stored for later plotting use, the number of points available for plotting was 6,884,408, about one-fifth the size of the cleaned data. Test values under 2 minutes were omitted, as quantities that small do not make logistical sense for train travel between these stops. Furthermore, for ease of viewing and understanding the plots, 1% of the predicted vs. actual delay data are displayed (45,873 coordinate pairs). It can be assumed that the randomly selected 1% is representative of the total.

3.1. Linear Regressor Model

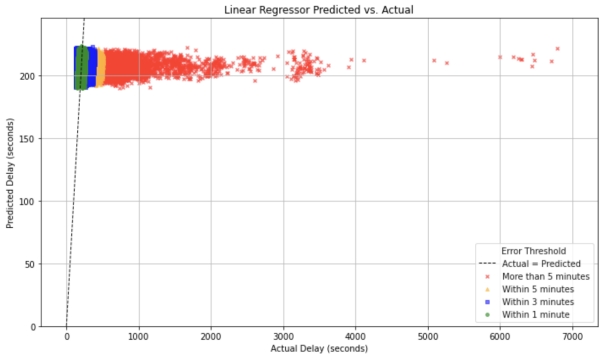

Figure 4: Linear Regressor Predicted vs. Actual Delay. For a wide domain of actual delays, the model predicts a relatively small range of predicted delays. The identity line (seen as dashed) represents the ideal path for points to fall upon, where the predicted and actual delay times are equal. It appears that much of the data fall around this line, but a significant fraction is also skewed to the right and under the line as well. For actual delays of more than 10 minutes and higher, the model predicts delays of less than 5 minutes. It appears that the model makes more accurate predictions for small values of actual delays, but for larger values it makes predictions with high errors. The errors do not seem to follow the trend of the identity line.

Metric	**Value**
Average Cross-validated R^2	0.0005
Test RMSE (seconds)	245.95

Table 2: Linear Regressor model performance gives a medium-high RMSE of about 245 seconds (4 minutes and 5 seconds). The 5 cross-validated R^2 scores appear due to the 5 folds the model uses for training and testing. The averaged R^2 value of 0.0005 is very close to 0, indicating the model explains almost none of the variance in the data better than the mean. Based on the R^2 and RMSE metrics for evaluation, it appears that the linear model does a poor job of predicting delay times, or that the data are not linear to begin with.

3.2. Decision Tree Model

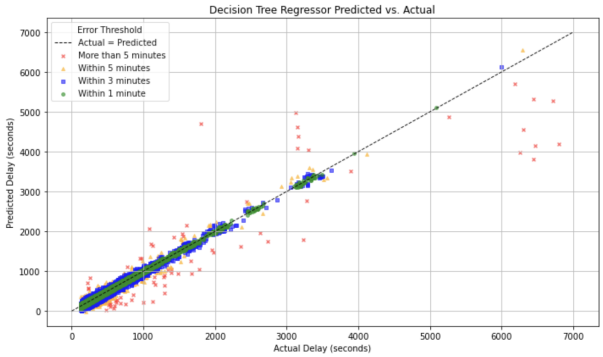

Figure 5: Decision Tree Predicted vs. Actual Delay. Most of the errors adhere closely to the identity line, indicating the model captures the general trend well. There are also some scattered values of more than 5 minute errors appearing at higher actual delays. This suggests that the model may be overfitting, as it is not predicting the error for long-delay journeys well. For the actual delays with high values, the predicted delays fall below the identity line, meaning that the model predicts these longer journeys will be shorter. It is also possible that the model is predicting accurately for normal case scenarios, and the discrepancies between long actual delays and shorter predicted delays are due to a train malfunction not captured within the scope of the data.

Metric	**Value**
Test R^2	0.9599
Test RMSE (seconds)	49.27

Table 3: Decision Tree model metrics indicate strong model performance. The R^2 value means that about 96% of the variance in delay times is explained by the model. The RMSE means that there is an average of about 49 seconds between predicted and actual values. Given that some of the actual values are upwards of 10 minutes, an RMSE of less than 1 minute is a low average error.

Parameter	Value
`min_samples_split`	10
`min_samples_leaf`	10
`max_features`	None
`max_depth`	None

Table 4: Best Parameters for Decision Tree of those defined in the model's hyperparameter space using a randomized search. The tree requires a minimum of 10 samples to split. A lower minimum than 10 may lead to overfitting, since splitting small subsets might capture noise rather than real trends in the data. The tree also requires at least 10 samples per leaf, limiting unnecessary complexity. With no maximum features, the tree considers all features before splitting. The tree can grow unbounded with no maximum depth. The strong regularization from `min_samples_split` and `min_samples_leaf` prevents the tree from growing excessively deep.

3.3. XGB Model

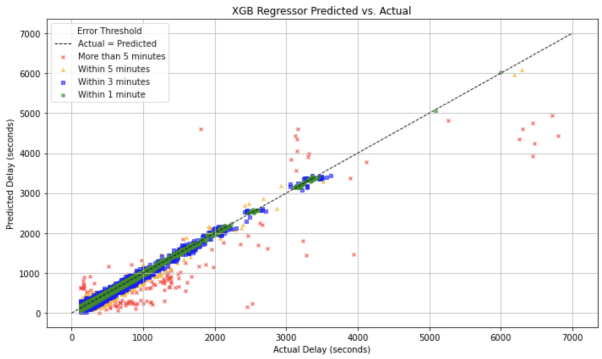

Figure 6: XGB Predicted vs. Actual Delay. The trend of the errors follows the identity line, meaning that the model is a good fit for the data. For many actual delays, both of large and small values, the model predicts shorter times, as the errors of more than 5 minutes fall below and to the right of the identity line. A possible explanation for more under- than over-predicting could be because the selected learning rate was 0.2. A relatively high learning rate like this can cause the model to make large updates early in the learning process, and biases toward lower values later in the training process even for longer actual delays.

Metric	Value
Test R^2	0.9502
Test RMSE (seconds)	54.90

Table 5: XGB model metrics indicate strong model performance. The R^2 value means that about 95% of the variance in delay times is explained by the model. The RMSE means that there is an average of about 55 seconds between predicted and actual values. Given that some of the actual values are upwards of 10 minutes, an RMSE of less than 1 minute is a low average error.

Parameter	Value
`subsample`	1.0
`n_estimators`	300
`max_depth`	10
`learning_rate`	0.2
`colsample_bytree`	1.0

Table 6: Best Parameters for XGB of those defined in the model's hyperparameter space using a randomized search. The subsample was 1.0, meaning the model used the entire dataset. The number of boosting rounds, `n_estimators`, was high, indicating that many rounds of new trees trained to correct the errors of existing trees were necessary due to the complexity of the data. The maximum depth was 10, suggesting that deeper trees were necessary as well. The learning rate selected was the highest of those available, meaning the model improved by more quickly adjusting its predictions in each step. Finally, the fraction of features the model uses for each tree (`colsample_bytree`) was 1.0, so the model benefitted when all available features were used.

3.4. Accuracy

Min.	LR	Dec. Tree	XGB
$\leq$1	35.995%	96.442%	95.508%
$\leq$3	88.826%	99.698%	99.542%
$\leq$5	93.514%	99.898%	99.805%
>5	100.000%	100.000%	100.000%

Table 7: Cumulative percentage accuracy of predicted delays within specified time thresholds by minute (Min.) for each model.

Metric	Value (seconds)
Arrival average	47
Arrival median	39
Departure average	647
Departure median	66
Linear Regressor RMSE	245.95
Decision Tree RMSE	49.27
XGB RMSE	54.90

Table 8: Comparison of uncleaned arrival and departure averages and medians and model RMSE values. The lowest RMSE value of about 49 seconds, from the Decision Tree, is closest to the arrival average difference of 47 seconds and is significantly lower than the departure average difference of 647 seconds (over 10 minutes).

4. Proposed Algorithm

For the reasons below, the Decision Tree Model is the proposed algorithm, should the Swiss rail authority choose to implement arrival time predictions using machine learning, or to incorporate the model into finding the likelihood of unexpected delays between target and actual times.

Of all three models, the Decision Tree ranks highest in percentages accurately predicted for all thresholds measured. It also has the best R^2 score and the lowest RMSE, performing within 3 seconds of the average difference between actual and target arrival times, and reduced error by a factor of 13 compared to the average difference between actual and target departure times.

While the differences in model performance between the Decision Tree and XGB regressor are not large, the Decision Tree model is significantly faster to run. The XGB regressor took not only more time, but also more computational power and memory in its execution. For a dataset as large as this, keeping in mind it only included the winter months, not the whole year, the XGB model is costly in terms of time and resources.

It is important to discuss the proposed algorithm's potential drawbacks as well. With the best performing Decision Tree having parameters with no maximum number of features and no maximum depth, the tree can grow very complex and deep. Since it can fit the training data nearly perfectly, it may also inadvertently fit noise. In the case of trains, noise can mean a broken sensor or an emergency causing a train line to stop for the day, giving unreliable data that should not be used for training. Even so, the benefits of accuracy performance in R^2 score, RMSE, and error thresholds, as well as relatively low computational cost, outweigh the risks of overfitting.

5. Conclusion

Train timetable accuracy is important to passengers for reliability, infrastructure for efficiency, and the rail company for consumer satisfaction. With the modern age of machine learning still in its early stages, applying these bourgeoning technologies to train systems has only recently begun. The objective of this dissertation is to provide one such model to predict travel duration times within the Swiss rail network using their publicly available data to train and test models, and to compare their efficacies.

This dissertation finds the Decision Tree most effective, compared to the Linear and XGB Regressors. Specifically, the Decision Tree with the parameters defined in Table 3.2. The criteria for best model were R^2 score, RMSE value, and accurate percentages at 1, 3, and 5 minute thresholds. The proposed tree considers all features and may grow unbounded, yet limits excessive complexity and depth.

These results demonstrate that a tree-based model, relatively simple compared to other possible approaches (see Section **??**), can still achieve strong predictive performance. This emphasizes the potential of data-driven modelling to improve operational insight and service reliability for modern rail systems.

5.1. Limitations

Throughout the dissertation writing process, I encountered several limitations. If not constrained by time and computational resources, I would have liked to delve deeper.

This dissertation makes use of data just from winter months, not the entire year. While I was able to use data from this period over multiple years, working with a whole year's worth of data over multiple years could have provided insight into seasonality trends. While outliers of errors greater than 5 minutes were discussed, a greater focus could be put on them in the future, investigating whether there were any clear patterns that might help mitigate large errors. Future research with more time and memory availability could try using a larger hyperparameter space, a grid search rather than a randomized search, and other models such as KNN and SVR.

There is still much to be discovered within field of modern machine learning. Future work, supported by greater computing power, may build on this approach to address similar challenges more effectively.

5.2. Extrapolations

This dissertation addresses the prediction times only of passenger trains in Switzerland, but the findings have the potential to be effective in other scenarios as well. The Decision Tree Regressor model proposed here, with its best performing parameters would likely perform well in other countries with similar times between stops, especially for short-distance passenger data. It is currently unknown by this author how the proposed model would perform for other types of transportation within the same original dataset (such as by ship or bus), or whether the model could be applied to other non-transportation time series data that also has predicted times. These are questions that may be explored in future research.

References

[1] Open Data Platform Mobility Switzerland. *Cookbook Brief Description*, 2025. https://opentransportdata.swiss/en/cookbook/historic-and-statistics-cookbook/actual-data

[2] Open Data Platform Mobility Switzerland. *Big Picture*, 2025. https://opentransportdata.swiss/en/cookbook/big-picture/

[3] Systemaufgaben Kindeninformation SKI. *Escalation Process*, 2025. https://www.oev-info.ch/de/datenmanagement/datenqualitaet/eskalationsprozess

[4] Open Data Platform Mobility Switzerland. *Actual Data*, 2025. https://data.opentransportdata.swiss/en/dataset/istdaten

[5] Open Data Platform Mobility Switzerland. *Cookbook Brief Description*, 2025. https://opentransportdata.swiss/en/cookbook/

[6] Open Data Platform Mobility Switzerland. *Actual Data Short Description*, 2025. https://opentransportdata.swiss/en/cookbook/historic-and-statistics-cookbook/actual-data/

[7] Open Data Platform Mobility Switzerland. *Ist-Daten Kurzbeschreibung*, 2025. https://opentransportdata.swiss/de/cookbook/historic-and-statistics-cookbook/actual-data/

[8] NumFOCUS, Inc. *pandas.DatetimeIndex.dayofweek*, 2024. https://pandas.pydata.org/docs/reference/api/pandas.DatetimeIndex.dayofweek.html

[9] European Passengers' Federation. *Punctuality Report of European Trains*, 2014. https://www.epf.eu/wp/10929-2/

[10] Agrawal, A., Kumar, V., Pandey, A., Khan, I. *An Application of Time Series Analysis for Weather Forecasting*, IJERA, 2012, Vol. 2(2), pp. 974–980. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=4a09fbc9f67613feca328370c95f75622d26f11d

[11] Zeroual, A., Harrou, F., Dairi, A., Sun, Y. *Deep Learning Methods for Forecasting COVID-19 Time-Series Data: A Comparative Study*, Chaos, Solitons & Fractals, 2020, Vol. 140, p. 110121. https://www.sciencedirect.com/science/article/pii/S096007792030518X

[12] scikit-learn developers. *LinearRegression*, 2025. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[13] scikit-learn developers. *DecisionTreeRegressor*, 2025. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

[14] scikit-learn developers. *Machine Learning in Python*, 2025. https://scikit-learn.org/stable/

[15] Iftikhar, H., Khan, F., Rodrigues, P. C., Alharbi, A. A., Allohibi, J. *Forecasting of Inflation Based on Univariate and Multivariate Time Series Models: An Empirical Application*, Mathematics, 2025, Vol. 13(7), p. 1121. https://doi.org/10.3390/math13071121

[16] Cheng, Y. H., Tsai, Y. C. *Train delay and perceived-wait time: passengers' perspective*, Transport Reviews, 2014, Vol. 34(6), pp. 710–729.

https://doi.org/10.1080/01441647.2014.975169

[17] Vafaei, S., Yaghini, M. *Online prediction of arrival and departure times in each station for passenger trains using machine learning methods*, Transportation Engineering, 2024, Vol. 16, p. 100250. https://www.sciencedirect.com/science/article/pii/S2666691X24000253

[18] XGBoost Developers. *XGBoost Python Package Documentation*, 2025. https://xgboost.readthedocs.io/en/latest/python/python_api.html

[19] Li, Z., Wen, C., Hu, R., Xu, C., Huang, P., Jiang, X. *Near-term train delay prediction in the Dutch railways network*, IJRT, 2021, Vol. 9(6), pp. 520–539.

[20] Shi, R., Xu, X., Li, J., Li, Y. *Prediction and analysis of train arrival delay based on XGBoost and Bayesian optimization*, Applied Soft Computing, 2021, Vol. 109, p. 107538.

[21] Yaghini, M., Khoshraftar, M. M., Seyedabadi, M. *Railway passenger train delay prediction via neural network model*, Journal of Advanced Transportation, 2013, Vol. 47(3), pp. 355–368.

[22] Murali, P., Dessouky, M., Ordóñez, F., Palmer, K. *A delay estimation technique for single and double-track railroads*, Transportation Research Part E, 2010, Vol. 46(4), pp. 483–495.

[23] Buijse, B. J., Reshadat, V., Enzing, O. W. *A Deep learning-based approach for train arrival time prediction*, IDEAL 2021, Springer, Manchester, UK, pp. 213–222.

[24] Kosolsombat, S., Limprasert, W. *Arrival time prediction and train tracking analysis*, PRICAI 2016 Workshops, Springer, Phuket, Thailand, pp. 170–177.

[25] Chen, Y., Rilett, L. R. *Train data collection and arrival time prediction system for highway–rail grade crossings*, Transportation Research Record, 2017, Vol. 2608(1), pp. 36–45.

[26] Presence Switzerland. *Transport*, 2024. https://www.aboutswitzerland.eda.admin.ch/en/transport

[27] Geotechnik Schweiz. *Gotthard Base Tunnel*, 2025. https://geotechnikschweiz.ch/?page_id=3972&lang=en

[28] Duranton, S., Audier, A., Hazan, J., Langhorn, M. P., Gauche, V. *The 2017 European Railway Performance Index*, Boston Consulting Group, 2017. https://www.bcg.com/publications/2017/transportation-travel-tourism-2017-european-railway-p

[29] Schubert, J., ABITZ.COM. *Zugfinder Live Map Europe*, 2024. https://www.zugfinder.net/en/livemap-europa

[30] Crobak, J. *parquet-python: A pure-python implementation of the Parquet format*, 2020. https://pypi.org/project/parquet/

[31] scikit-learn developers. *LabelEncoder — scikit-learn 1.6.1 documentation*, 2025. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[32] scikit-learn developers. *scikit-learn model selection cross validate 1.6.1 documentation*, 2025. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_validate.html

[33] Net Zero Nation. *Benefits of Public Transport*, 2025. https://netzeronation.scot/take-action/travel-less-car/benefits-public-transport