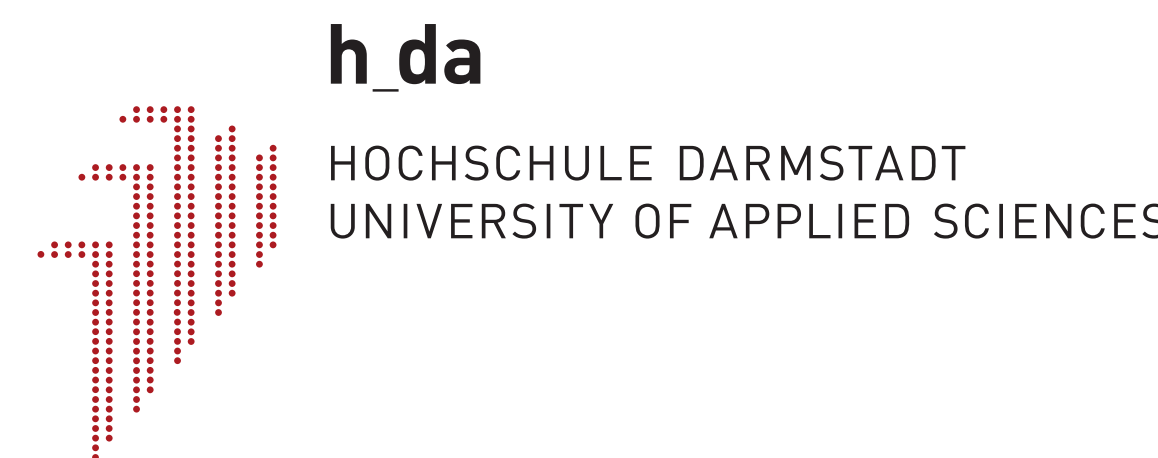


Machine Learning-Based Runtime Prediction for CAD Data Migration: A Case Study

Katharina Litzinger

Supervisors: Prof. Dr. Timo Schürg, Prof. Dr. Melanie Siegel

Hochschule Darmstadt – Fachbereich Mathematik und Naturwissenschaften & Fachbereich Informatik



Summary

This Master's thesis addresses the challenge of runtime prediction for CAD data migration using machine learning techniques. The research pursues two primary objectives: predicting the migration runtime either in parts or as a whole, and conducting a comprehensive analysis of feature importance. The study implemented a two-stage cascading model stack to reduce skewness of the target variable. predict the runtime of a CAD-file migration task. XGBoost models proved to be robust and efficient in both target domains, classification and regression. The Forward sequential feature selector (Forward SFS) achieved the best mean score most often during feature selection. Predictions within individual stages still exhibit challenges, particularly regarding non-outliers. The inclusion or exclusion of correlated features did not yield a universally superior approach. Instead, it proved to be highly context-dependent while improving specific tasks, simultaneously exposing a pronounced seed sensitivity due to the limited number of samples.

Motivation

The market for Computer Aided Design (CAD) systems is vast and diverse. Much like popular word processing programmes such as Microsoft Word, Apple Pages and Google Docs, the main problem with CAD files is that files from different manufacturers are generally not fully compatible with one another [Max02], [Azu93, p.1]. Despite the growing demand for efficient data exchange, the overall migration process between formats like open source to commercial format or from vendor to vendor is sparsely documented in research. This might be because vendor specific CAD-file formats are often considered intellectual property and therefore rarely appear in open source data sets or research papers [PS18, p. 55]. This study aims to help address the research gap by evaluating metadata which is gathered during a specific conversion process. Machine Learning (ML) is applied to predict the migration runtime and assess the influence of individual features as well as the feasibility of the presented approach. This thesis also seeks to deepen the technical understanding of CAD file migration and provide foundation for future research in this area. The presented study is structured around the following three research questions:

- R1 Can the duration of a migration process be predicted, either in parts or as a whole?
- R2 Which features are influencing runtime the most?
- R3 What additional data from the source or target system could be collected to potentially improve the migration?

Data & Preprocessing

The dataset stems from a training migration and comprises, after preprocessing, 1,672 samples across 111 columns, with a moderate class imbalance of 0.05 addressed through cost-sensitive learning. Restricting the data to fields provided by the customer reduces the column count to 105. For both datasets, the inclusion versus exclusion of highly correlated features was evaluated. For data cleaning and feature engineering, missing values were addressed via additional indicator columns. A custom frequency encoder extracted information from highly individual categorical columns while reducing their dimensionality. Numeric columns were transformed using Yeo-Johnson and subsequently scaled. Additional features were engineered from existing columns as well as from log-file extraction. The data set introduced a highly right skewed target variable (Figure 1). Evaluation results showed, that a direct regression model cannot handle short and long durations simultaneously. Binning the target variable also brings no superior performance and lessens granularity, illustrating that classification alone also is not sufficient for the task.

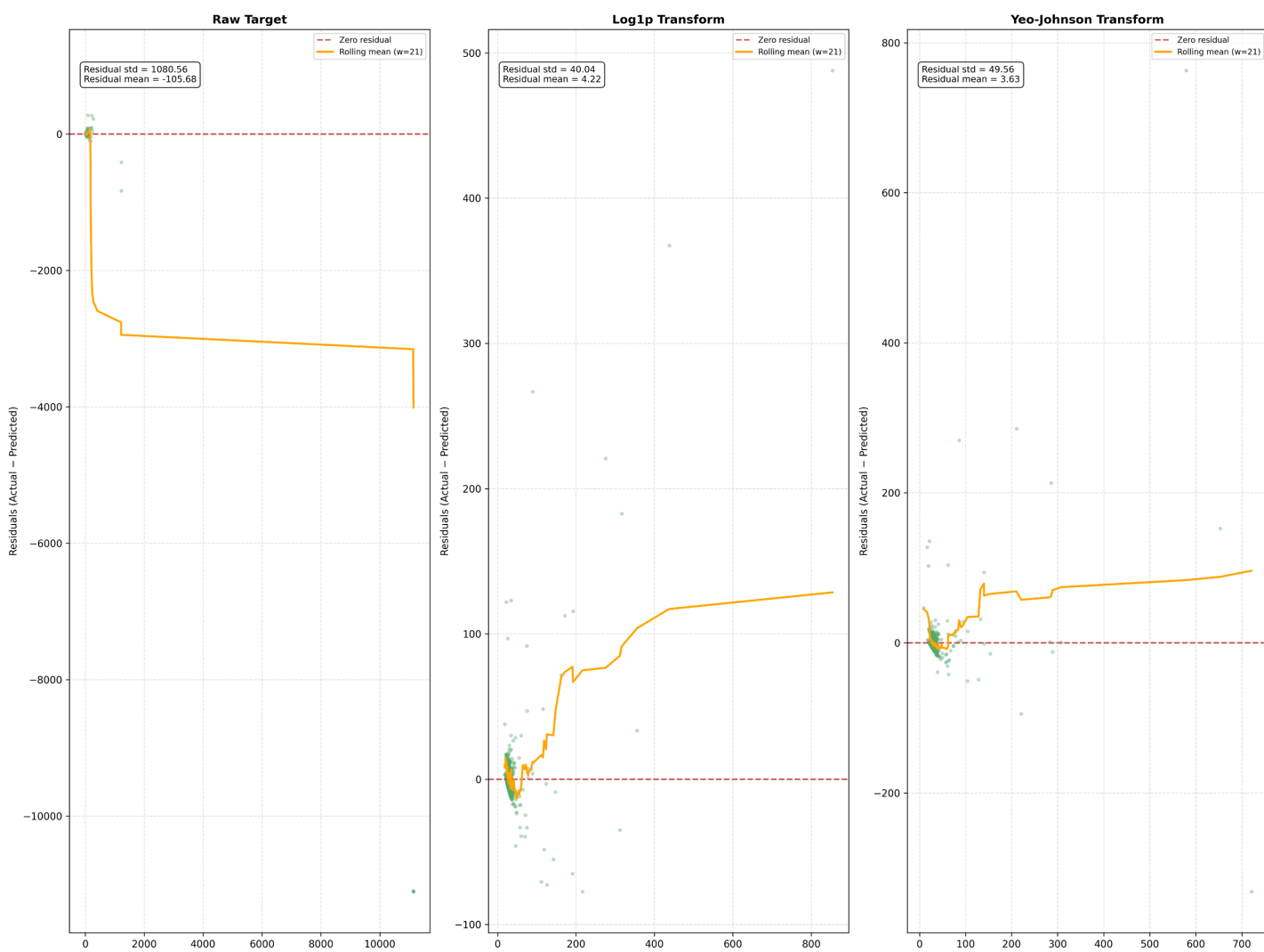


Figure 1: Prediction strength and residual behaviour of raw vs transformed and power-transformed conversion_processduration target variable showing heteroscedasticity behaviour

Study Development and Methodology

To address this distributional challenge, a two-stage approach was developed. The concept is to reformulate the skewed regression problem as a sequential classification and regression task, isolating outlier in a pre-classification stage to reduce skewness in the following steps. Consequently this approach was realized as a two-stage cascading model stack, Figure 2

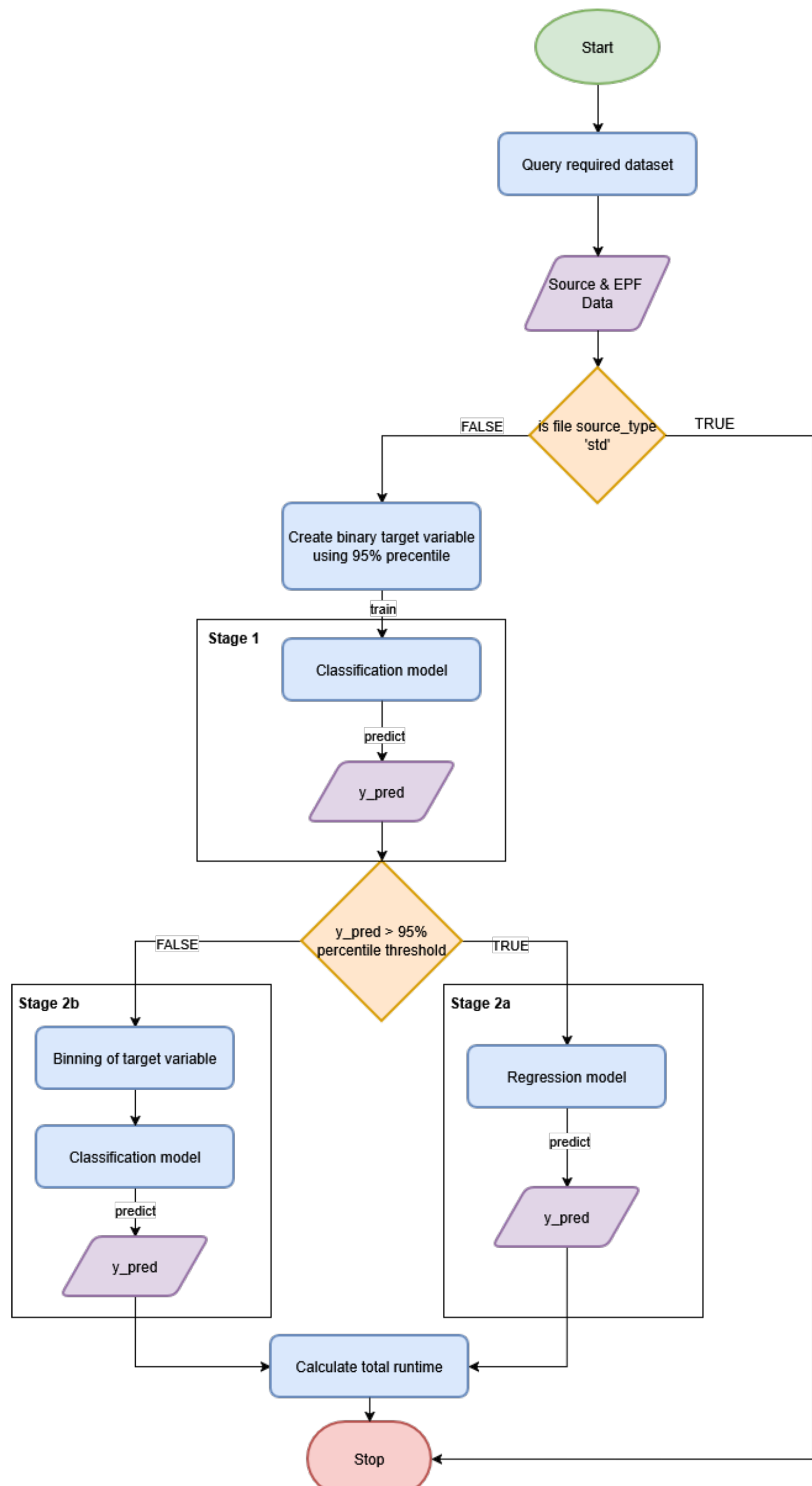


Figure 2: Architecture of the cascading model stack for total runtime prediction. The flow chart illustrates the sequential prediction process, where the output of each model is denoted as y_{pred} (own graphic).

Stage 1: A binary classifier predicts whether a file's conversion process duration will exceed the 95th percentile of the training distribution. These files are labelled outliers. This specific threshold was selected to improve predictive performance while maintaining sufficient training samples in the outlier class and to identify files with a potentially long runtime early in the pipeline. To select the best model a comprehensive comparison between models is performed with stratified k-fold cross-validation as well as MCC and recall as evaluation metrics. To improve the selected model performance, hyperparameter optimization is performed. To potentially increase performance further, different approaches for feature selection are evaluated. The final fine-tuned model generates out-of-folds prediction using **cross_val_predict** to avoid data leakage between stages. Potential misrouted entries are not corrected post-hoc to reflect real-world deployment conditions more accurately.

Stage 2a - Outlier Regression: Files classified as outliers are processed by a regression model that predicts exact duration. This specialized model can focus on the tail distribution without being dominated by the majority class. Here the same steps as above are used: Comprehensive model selection, hyperparameter tuning and feature selection. All of these steps are modified to suit a regression model and evaluated by a k-fold cv with R^2 , MAE, RMSE and MSE.

Stage 2b - Non-Outlier Approximation: To predict the non-outlier data points different approaches will be studied. Regression, Semantic binning, Quantile-based binning, Binary binning, Three-class binning as well as statically assigning runtimes at different ratios. All approaches are evaluated using k-fold cv and R^2 , MAE and RMSE for regression and MCC and (macro)-recall for binary and multiclass classification.

Obj.	Model Comparison	Hyperparameter Optimization	Feature Selection	Eval.
Clf.	XGBoost, Logistic Regression, SVM, Random Forest	Random search, Bayesian optimization	Foward SFS, RFECV, Lasso, threshold-based (mean, median, 0.01*max), percentile-based (25, 50, 75, 90)	MCC, (macro) recall
Reg.	XGBoost (MAE), Boost (MSE), Linear Regression, Random Forest Regressor, Quantile Regression	Random search, Bayesian optimization	Foward SFS, RFECV, Lasso, threshold-based (mean, median, 0.01*max), percentile-based (25, 50, 75, 90)	R^2 , MAE, RMSE

Table 1: Summary of used models and methodologies during the model stack development, divided into model comparison, hyperparameter optimization, feature selection and evaluation metrics.

Results

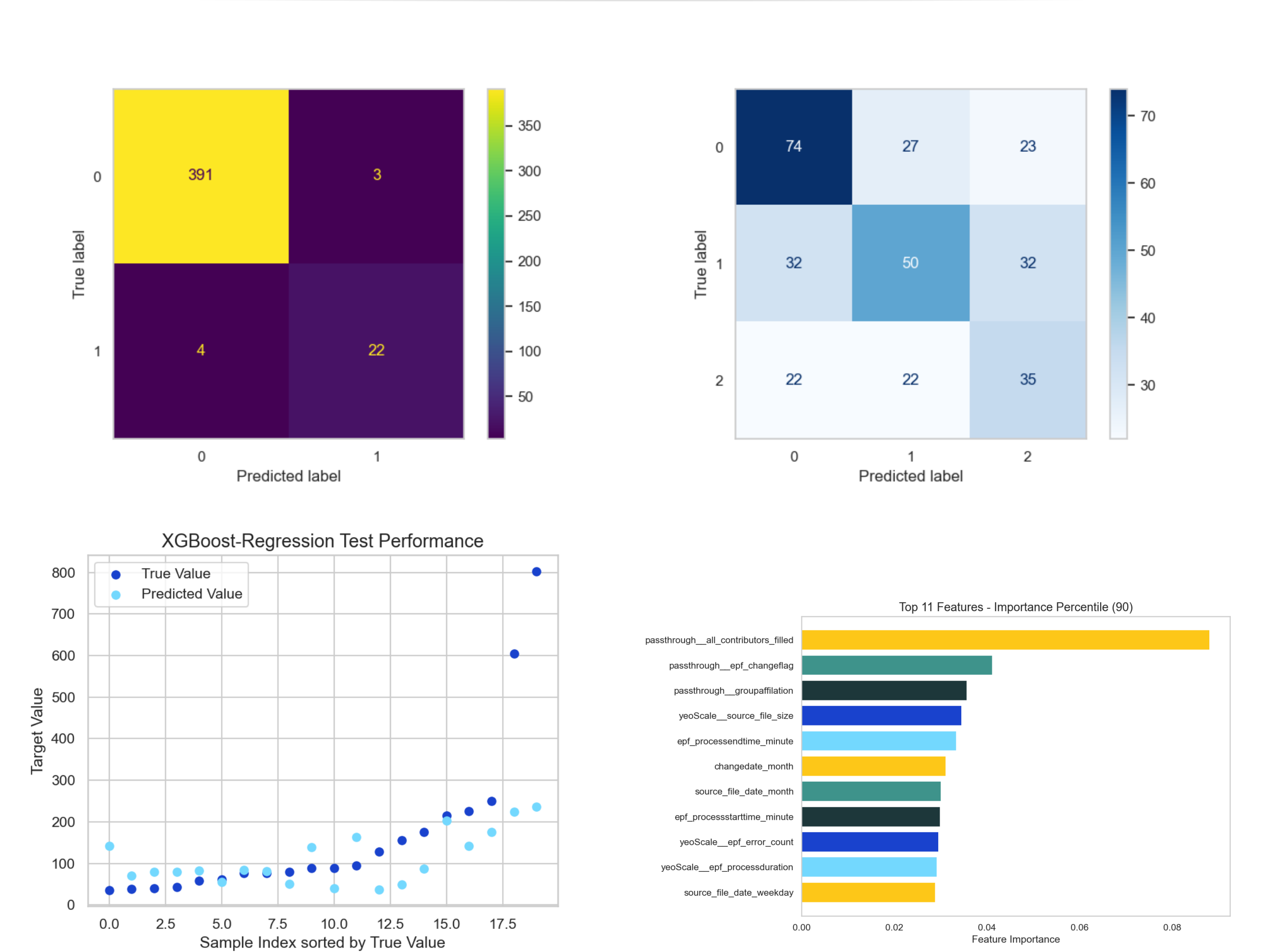


Table 2: Excerpt of visualized results of the conversion_processduration excl. correlation

The **conversion_processduration** excluding correlated features processes on the outlier branch (Stage 2a) 20 test samples and achieves a MAE of 92.85s, resulting in an absolute error of 1120.93s (33.55%). The non-outlier branch (Stage 2b) processes 317 test samples with a MAE of 8.62s and an absolute error of 56s (0.57%). Combined, the pipeline yields a test MAE of 13.62s with a total absolute error of 1064.93s and a relative error of 8.04%. On the training set, the combined relative error amounts to 24.47%. The feature selection reduced the number of features of Stage 1 by 74.3% and by 89.5% for each Stage 2 branch. Regarding all samples routed within the cascading model stack, 25 were misrouted as false negatives (1.49%), 37 (2.21%) false positives. Combined this results in a total of 62 misrouted files (3.69%). Stage 2a, of the customer only predicting **total_processduration** excluding correlated features, predicts 3422.4s which is 752.60s below the true runtime of 4175.0s. This means a relative error of 18.03% and a MAE of 62.65s. Stage 2b model predicts 15522 overestimating the actual runtime by 129s with a relative error of 0.84% and MAE of 9.91s. On the training set the combined relative error comes to 20.38% whereas the test set shows a relative error of 3.19%. On the MAE this is a reduction from 25.11s down to 13.84s. The feature selection leads to feature reduction in all stages. In Stage 1 the 77 features are reduced by 66.2% to 26 features, Stage 2a by 83.1% to 13 features and the most reduction of 89.6% in Stage 2b to 8 features. The total impact of misrouted files is for false negative samples 0.83% (14 samples), for false positives 3.28% (55 samples) totalling 4.11% (69 files).

In table 2 on the left top side the confusion matrix of Stage 1 on the test set is presented, next to it the confusion matrix of Stage 2a is presented. On the down left side, a scatter plot displays the distance between true and predicted runtimes for Stage 2a alongside the feature selection results of Stage 2a. This gives a general overview of the results achieved by using conversion data.

Conclusion

The results demonstrate that the proposed architecture with XGBoost and cost-sensitive learning, is a valid approach for this domain. The model successfully learned underlying patterns for mid-range runtimes and effectively minimizing the cost through recall was prioritized for long running files. Through a comprehensive feature importance comparison, this study showed that high-level metadata, especially the initial file size, assembly type and earlier runtimes propose a stronger role than the number structural dependencies. Furthermore, introducing domain knowledge manually significantly enhanced the model's predictive strength. The evaluation, however, shows that extreme outliers are difficult to predict solely through file data. A potential cause is the dependency on external factors such as network load, etc.. Additionally, the model's performance is limited by the size of training and test data, resulting in seed-sensitivity during cross-validation. The instability of the prediction also indicate that the most influential features might not yet be acquired. In conclusion, the model cannot generalize on unseen live migrations due to data scarcity, but this study successfully established a robust proof of concept. It proves that ML can be utilized to extract valuable operation insights from migration databases.

Quellen

- [Azu93] Azu, Charles C., Jr. "An application protocol for CAD to CAD transfer of electronic information". In: *NASA, Washington, Technology 2002: The Third National Technology Transfer Conference and Exposition, Volume 2* (1993). URL: <https://ntrs.nasa.gov/citations/19930012965>.
- [Max02] Max Planck Digital Library. *File Formats – Research Data Management*. 2026-06-02. URL: <https://rdm.mpg.de/before-research/file-formats/> (visited on 06/02/2026).
- [PS18] Alain Pfouga and Josip Stjepandić. "Leveraging 3D geometric knowledge in the product lifecycle based on industrial standards". In: *Journal of Computational Design and Engineering* 5.1 (2018), pp. 54–67. ISSN: 2288-5048. DOI: 10.1016/j.jcde.2017.11.002. URL: <https://www.sciencedirect.com/science/article/pii/S2288430017300441>.