BIP Machine Learning for Data Science

-Project Documentation-

Group B - ML4DS

February 2, 2025

Felipe Merenda Izidorio, Guilherme Quintero Lorenzi, Jost Nickel, Néstor Rubén Delgado Feliciano

Hochschule Bremen | Faculty 4: School of Electrical Engineering and Computer Science

---

Examiner: Prof. Dr. Uta Bohnebeck

Contents

List of Tables

List of Figures

Abbreviations

EDA	exploratory data analysis
RMSE	Root Mean Squared Error
RF	Random Forest
SVM	Support Vector Machine

1. Introduction

The automotive industry is highly competitive, and car dealerships face significant challenges in optimizing their operations to maintain profitability and customer satisfaction. One critical aspect of dealership efficiency is minimizing the time vehicles spend unsold at the dealership, referred to as the 'laid-up time'. A long laid-up time can lead to increased inventory costs, reduced profitability, and inefficiencies in dealership operations. This project aims to address this challenge by developing a predictive model to estimate the laid-up time of vehicles at the time of purchase. By leveraging historical sales data and advanced machine learning techniques, the project not only seeks to provide accurate predictions but also to identify key features that influence laid-up time. Additionally, it aims to offer actionable insights into which vehicle specifications are associated with shorter laid-up times.

The dataset used in this study, provided by Emil Frey, contains over 140,000 records of vehicle sales spanning a 10-year period, with more than 100 features. A training dataset of approximately 100,000 records includes the laid-up time, while a test dataset of 40,000 records lacks this target variable and is used for evaluation purposes. The performance of the model will be assessed using the Root Mean Squared Error (RMSE) metric, ensuring a robust evaluation of predictive accuracy.

Predictive modeling has become a crucial tool in the automotive industry, helping businesses make informed decisions and streamline operations. As noted by Doe et al. (2023), data-driven approaches provide valuable insights for inventory management and demand forecasting [1].

This report presents a comprehensive end-to-end approach, from data preprocessing and exploratory data analysis (EDA) to model development, validation, and deployment. Additionally, it explores the key attributes driving laid-up time and provides recommendations for dealership inventory optimization.

The complete code, models, and other resources for this project are available in the Git repository at the following link: `https://github.com/JONICK277/ML/.`

2. Problem Description

The goal of this project is to build a predictive model that estimates the laid-up time for vehicles using historical sales data provided by Fa. Emil Frey. In addition to building the model, participants are encouraged to analyze the dataset to answer key analytical questions:

- What features or attributes influence the laid-up time at car dealerships? Which features have the most significant impact on this duration?
- What types of vehicles (based on their specifications) should be prioritized for purchase to minimize laid-up time at the dealership?

Dataset Details

The dataset provided by Fa. Emil Frey, with a 10-year history of vehicle sales in/ its dealerships. It contains information on more than 140,000 car sales, described by over 100 features. The dataset is

split into two subsets:

- **Training Dataset:** Includes approximately 100,000 records, each containing all feature data along with the target variable (`LAID_UP_TIME`).
- **Test Dataset:** Consists of approximately 40,000 records where the target variable (`LAID_UP_TIME`) is absent and needs to be predicted.

Each of the different features contained in the data sets is explained in the appendix A.1. The data files are available through the Aulis platform in the subfolder named `Data`.

Deliverables

Participants are required to submit the following:

1. A trained predictive model.
2. A file containing the predictions for the test dataset. This file must include two columns:
   - `CHASSIS_NUMBER`
   - `LAID_UP_TIME`

Model Evaluation

The performance of the submitted models will be evaluated by comparing the predicted `LAID_UP_TIME` values with the true values. This evaluation will be carried out by the instructors based on the RMSE metric. The goal is to develop a model with the highest accuracy and to provide insights into the factors influencing the laid-up time.

3. Preprocessing

3.1. Data Overview

The dataset used for this project is divided into two parts: a training dataset and a test dataset. Below are the details regarding the size and structure of these datasets:

- **Training Dataset:** Contains 99,071 rows and 106 columns.
- **Test Dataset:** Contains 42,425 rows and 106 columns.

The dataset includes a variety of data types, as summarized below:

- **Object:** 59 columns contain categorical or string-type information. Examples include "OFFICE", "VEHICLE_TYPE" and "COLOR".
- **Float64:** 42 columns contain numerical data, such as "MILEAGE", "PERMITTED_TOTAL_WEIGHT" and "YEAR_CONSTRUCTION".

- **Datetime64:** 5 columns store date and time values, including "PURCHASE_DATE" and "PURCHASE_BOOKING_DATE".

Despite its rich structure, the dataset presents potential challenges that must be addressed during the pre-processing phase. Missing values are likely to be present because of the extensive number of features, which can affect the model's performance if not handled properly. Numerical features may contain outliers, potentially distorting the analysis and leading to overfitting in machine learning models. In addition, some object-type features might have inconsistent data formats or values, which requires normalization for effective analysis.

The dataset's high dimensionality, with 106 features, poses another concern. The inclusion of many variables can introduce multicollinearity and increase the complexity of the model, making it difficult to interpret.

To address the issue of irrelevant or non-informative features, we implemented a feature selection technique by removing columns with a standard deviation of zero. These columns do not provide variability and therefore don't contribute useful information to the machine learning model. After applying this approach, the number of features was reduced, resulting in a more simplified data set.

A bar plot visualization was generated to illustrate the distribution of standard deviations for numerical columns. In the plot, columns with a zero standard deviation are highlighted with red, indicating the features that were removed during this preprocessing step. The graphic below (Figure 1) clearly shows which features were eliminated, supporting a cleaner and more informative data set for subsequent analysis.

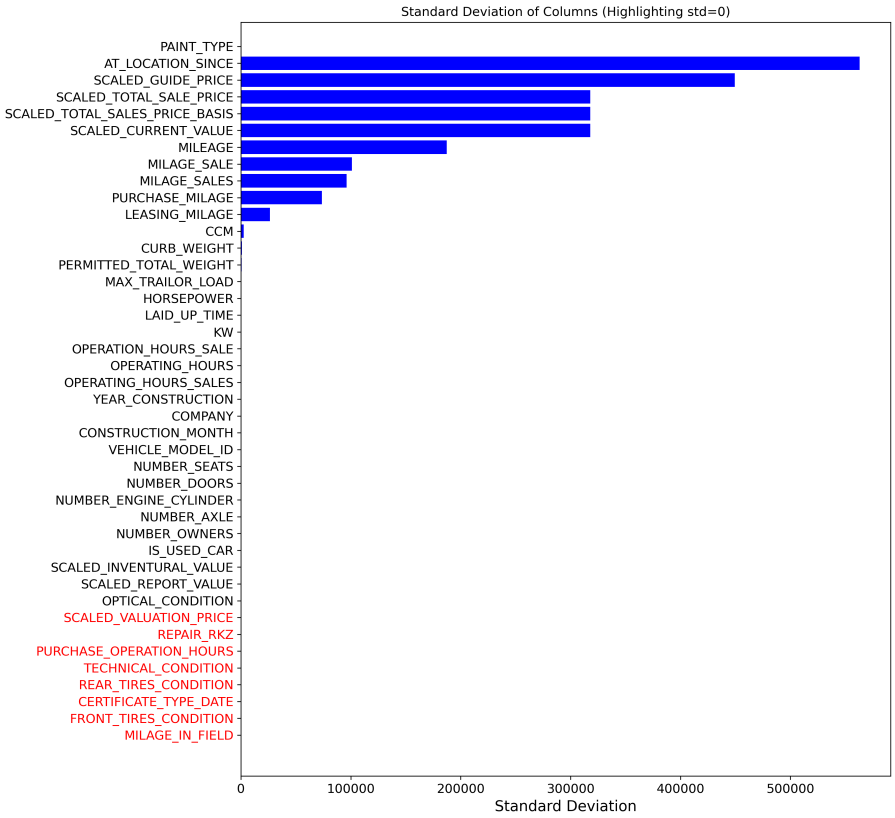

Figure 1: Standard Deviation of Numerical columns.

After excluding the features with a standard deviation of zero, the dataset was left with 98 features in total. After this, a search was made for duplicate instances in the training dataset, and 199 instances were discovered and excluded.

3.2. Missing Values

Missing values are a common issue in real-world datasets and can significantly affect the performance of machine learning models if not properly handled. To better understand the extent of missing data in the dataset, a heatmap (Figure 2) was generated to visually highlight the locations of missing values. This allows for a quick overview of which columns and rows have missing data, providing a clearer understanding of how to handle them. The heatmap highlights columns and rows with missing data, where each cell is colored based on the presence of missing values. The darker shades indicate the presence of data, and the lighter shades represent the missing values. This visualization provides a clear understanding of where the missing values are concentrated in the data set.

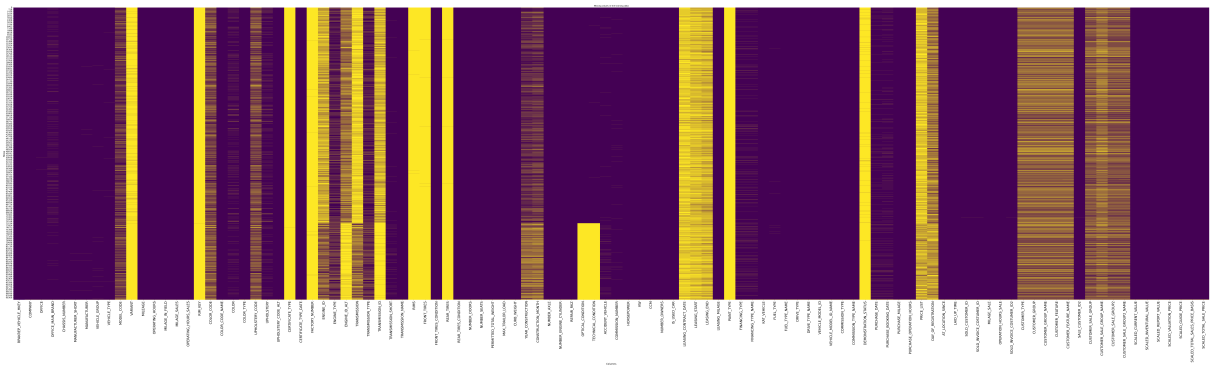

Figure 2: Heatmap of Missing Values in the dataset.

Based on the analysis of the missing data from the heatmap, we applied the following strategies for pre-processing:

- **Exclusion of columns with more than 70% missing values:** these columns were considered to contain too much missing information and were removed from the dataset. This step helps to eliminate features that would provide very little useful data for analysis.
- **Exclusion of rows with more than 50% missing values:** these rows were removed to prevent them from distorting the analysis. By removing these rows, we ensured that the data set retained enough information to train and test the model.

By applying the first strategy, the number of features was reduced to 69, making it easier to analyze these columns to identify the most relevant ones. In the second strategy, 51 instances were removed from the training set, providing a more consistent basis for filling in numerical values by average, if necessary.

3.3. Removal of irrelevant or redundant columns

Another important step in the pre-processing phase was to remove irrelevant and redundant columns. These columns, although present in the dataset, do not contribute significant information to the training of the model and can even introduce noise or complexity.

The features removed included unique identifiers such as "CHASSIS_NUMBER" and "RPAKREP_VEHICLE_HKEY", which only served to distinguish individual vehicles without providing predictive insights. Similarly, features like "MANUFACTURER_SHORT" and "MODEL_CODE" were removed because they represented redundant or overly specific vehicle details already captured by broader features within the dataset.

Several columns related to vehicle operating times, including "OPERATING_HOURS", "OPERATING_HOURS_SALES", and "OPERATION_HOURS_SALE", were excluded due to their limited or overlapping value with other existing features. The "COLOR_CODE_NAME" column was also removed as it provided redundant information already captured in the "COLOR" column, while

"TRANSMISSION_SHORT" was deemed unnecessary given the presence of more detailed transmission descriptors. The "OPTICAL_CONDITION" column, after evaluation, was found to offer no significant predictive value in the context of the project and was thus eliminated.

Furthermore, the "COMMISSION_NUMBER" column was excluded for being a unique identifier without any modeling relevance. Features such as "FINANCING_TYPE" and "KAT_VEHICLE" were considered redundant or non-contributive, while "FUEL_TYPE_NAME", "DRIVE_TYPE_NAME", "VEHICLE_MODEL_ID_NAME", and "COMMISSION_TYPE_NAME" were found to be repetitive, with their information already encoded in broader features. Lastly, customer-specific identifiers like "SOLD_CUSTOMER_ID", "SOLD_INVOICE_COSTUMER_ID", "SOLDICE_COSTUMER_ID2", and "SALE_CUSTOMER_ID2" were also removed, as they did not provide meaningful insights for the intended analysis.

By removing these columns, the dataset was refined to reduce noise and redundancy, simplifying the feature space and improving the suitability of the dataset for developing the machine learning model.

3.4. Color Handling

The dataset contains a column labeled "COLOR" with a significant number of unique values. These values represent different colors of vehicles, but the column presents several challenges: there are numerous unique color entries, and many of them are recorded in different languages. This variation makes it difficult to standardize and analyze the color data effectively.

To address this issue, an approach to normalize the color values was implemented by creating a dictionary that includes various color names along with their translations across different languages. This dictionary serves as a mapping tool that allows us to standardize the color values into a common set of categories. For example, color names such as "Blanco" (Spanish), "Weiß" (German), and "White" (English) were all mapped to a unified value of "White". In addition, if the color value was not present in the dictionary, it was replaced by “OTHER”.

By applying this dictionary to the "COLOR" column, we were able to replace the original color entries with the corresponding values from the dictionary. Afterward, we aggregated the results into broader categories, ensuring that the color information was both standardized and simplified. This approach (Figure 3) reduced the number of unique values, making the dataset easier to handle and enabling more effective analysis and modeling.

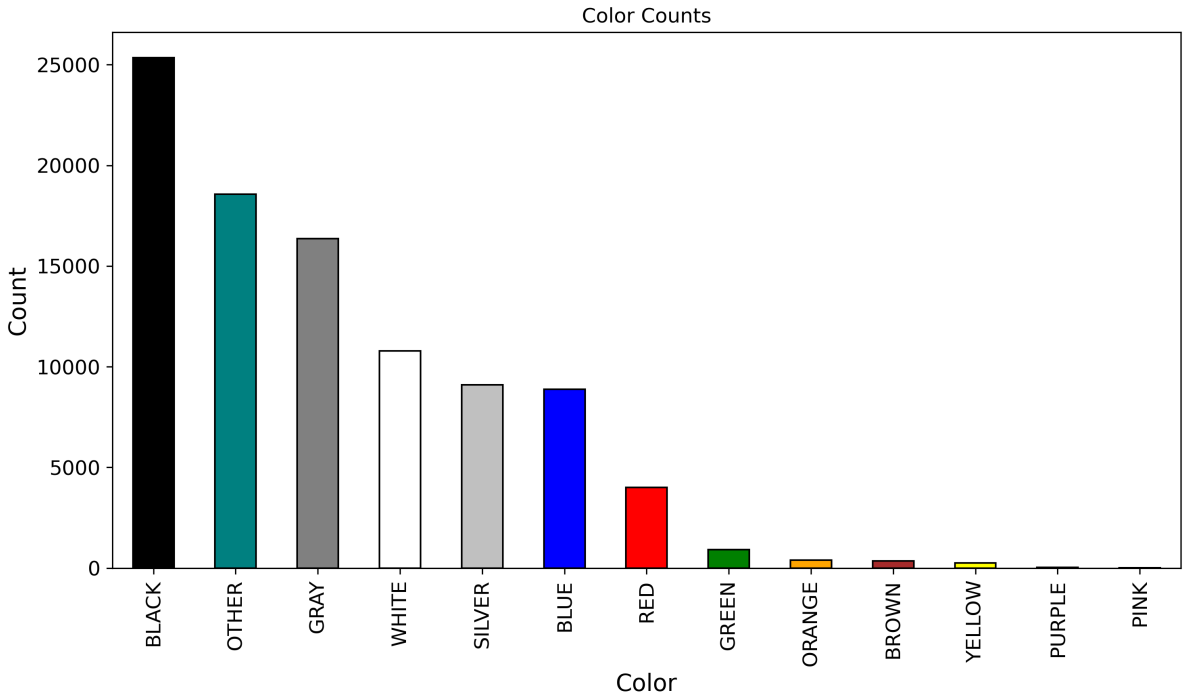

Figure 3: Color counting after color handling.

3.5. Feature Engineering from Date Columns

To enrich the dataset, we extracted new features from the existing date columns. The following features were created:

- **Day**: The day of the month.
- **Weekday**: The day of the week (e.g., Monday, Tuesday).
- **Month**: The month of the year.
- **Year**: The year.

This transformation increased the number of columns from **48 to 54**, providing additional temporal information that could improve the model's predictive performance.

3.6. Outliers Handling

To mitigate the influence of extreme values, we applied the **Winsorizing** technique. This method replaces extreme values with the closest acceptable values within a predefined range. In our case, we selected the 0.05 quantile as the threshold to define the acceptable range for the data.

To exemplify this, we present boxplots (Figure 4) of the variable **CURB_WEIGHT** before and after applying Winsorizing.

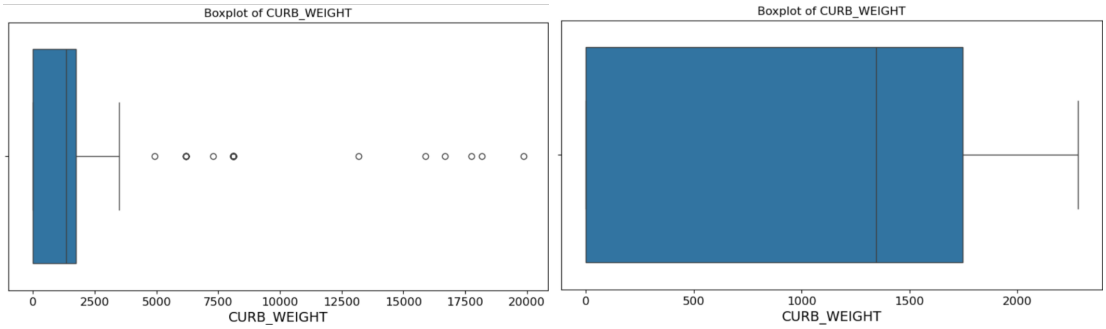

Figure 4: Boxplot of CURB_WEIGHT before and after Winsorizing.

Curb weight (or kerb weight) refers to the weight of a vehicle with all its essential fluids (oil, coolant, fuel at the minimum operating level, etc.) but without passengers or cargo. In the dataset we analyzed, which consists of cars, clear outliers are evident, as some values exceed 7 tons—weights that are unrealistic for standard cars. After applying the Winsorizing technique, these outliers were replaced with more realistic values based on the overall data distribution, using the previously mentioned threshold (0.05 quantile). This adjustment ensures the data better reflects the typical weight range for vehicles.

3.7. Filling Missing Values

Missing values were addressed as follows:

- **Categorical Columns**: Replaced with the label *"Missing"*.
- **Numerical Columns**: Filled with the column mean.

This ensured data completeness while preserving the dataset's structure.

3.8. Testing Different Encoding Methods

To convert categorical variables into numerical representations, we tested the following encoding techniques:

1. **One-Hot Encoding**: Our initial approach, which resulted in a significant increase in the number of columns due to the high cardinality of some categorical variables. This led to computational challenges, as our system struggled to handle the expanded dataset.
2. **Label Encoding**: This method performed better than One-Hot Encoding, as it did not drastically increase the number of columns.
3. **Frequency Encoding**: Showed similar performance to Label Encoding.

4. **Target Encoding**: Achieved the best results in terms of RMSE and correlation with the target variable (`LAID_UP_TIME`).

Based on the results, **Target Encoding** was applied to all categorical columns as mixing methods did not yield notable improvements.

To compare the performance of different encoding types, we rely on the RMSE (Table 1) and the correlation between the encoded categorical columns and the target variable `LAID_UP_TIME` (Figure 5).

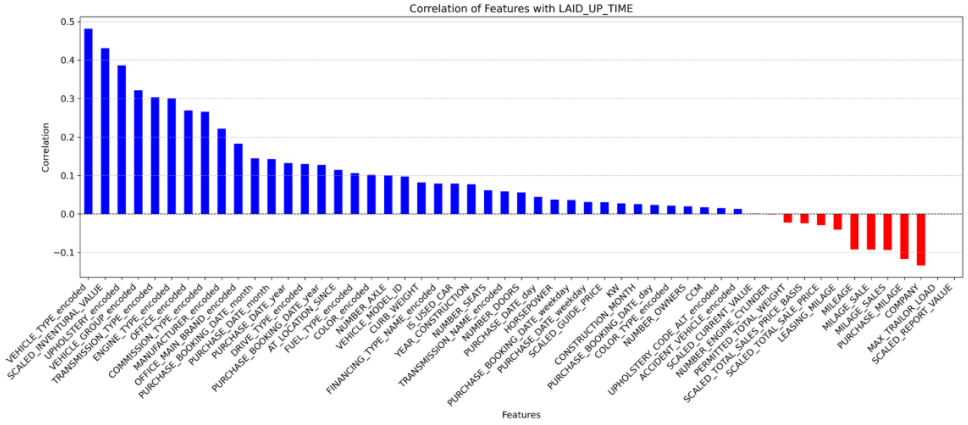

Figure 5: Correlation between encoded categorical variables and the target variable.

Table 1: Comparison of RMSE for different encoding methods.

Encoding Method	RMSE
One-Hot Encoding	-
Label Encoding	41.193
Frequency Encoding	40.628
Target Encoding	**38.183**

3.9. Deleting Low-Correlated Columns

To enhance efficiency, we removed features with low correlation to the target variable (`LAID_UP_TIME`). This step reduced the number of columns from **52 to 33** (Figure 6, improving computational performance while retaining relevant features.

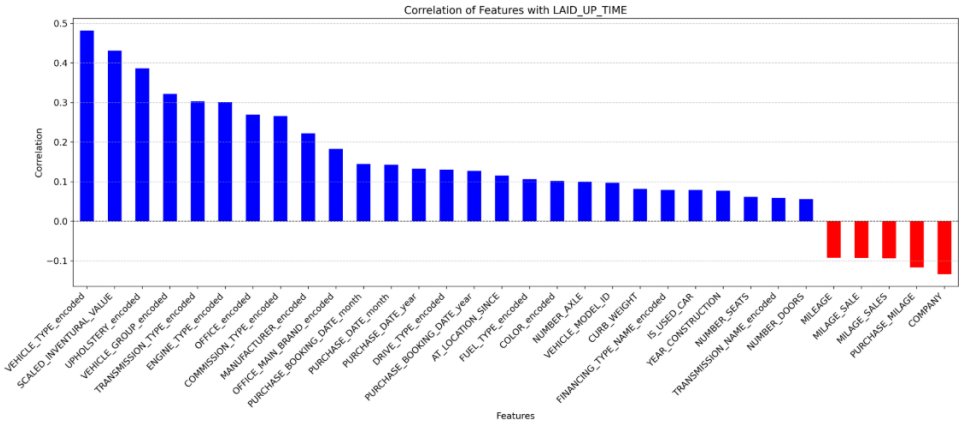

Figure 6: Correlation of remaining features after low-correlation columns were removed.

3.10. Summary of Feature Evolution

The following graph (Figure 7) illustrates the progression of the number of features throughout the preprocessing phase. Although the feature count increased due to date feature extraction, subsequent steps significantly reduced it.

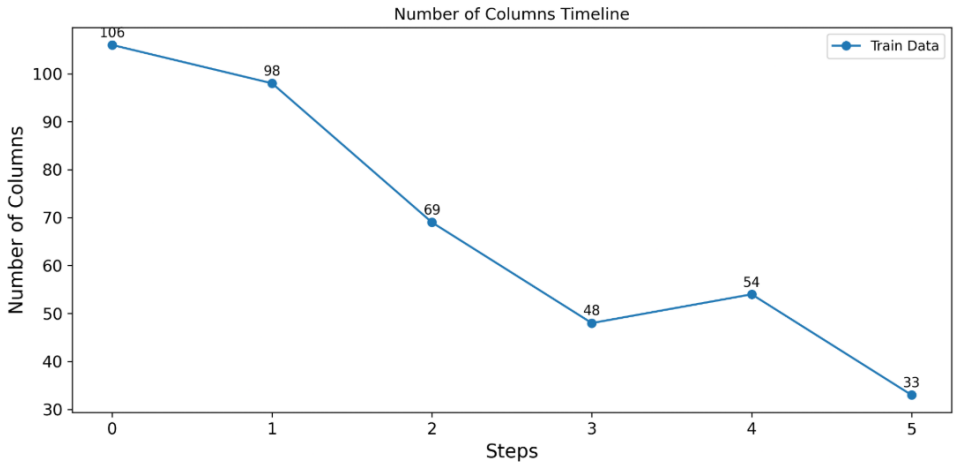

Figure 7: Evolution of the number of features during preprocessing.

4. Method Selection

In this chapter, we describe the approach taken to select and optimize the models used to predict `LAID_UP_TIME` at the dealership. Given the dataset's complexity, which includes a mix of numerical, categorical, and temporal data, various machine learning models were evaluated to determine the most effective approach.

4.1. Initial Model Exploration

During the initial phase, we experimented with several regression models to establish baseline performance and understand their suitability for the task:

- **Decision Tree Regressor:** A simple and interpretable model but highly prone to overfitting due to its reliance on splitting rules.
- **Support Vector Machine (SVM):** Capable of handling complex relationships but computationally expensive and sensitive to hyperparameter choices.
- **Neural Networks:** Despite their ability to model complex patterns, they require extensive hyperparameter tuning and large amounts of data to generalize well.

While these models provided insight into the structure of the problem, they were ultimately found to be suboptimal due to their sensitivity to noisy data, lack of generalization, and computational inefficiencies.

4.2. Selection of Ensemble Learning Methods

Based on the limitations observed in the initial models, we turned to ensemble learning methods, which are known for their robustness, interpretability, and ability to handle structured data effectively. The following ensemble methods were selected:

- **Random Forest Regressor:** An ensemble of decision trees that reduces variance through bagging (bootstrap aggregation). It also provides useful feature importance scores.
- **XGBoost:** An optimized gradient boosting technique that performs well on structured data. Its ability to handle missing values and incorporate regularization makes it suitable for our dataset.
- **Gradient Boosting Regressor:** Similar to XGBoost, but with an emphasis on reducing bias through iterative boosting while maintaining interpretability.

These models were chosen based on their ability to:

- Handle missing data and categorical variables effectively.
- Provide feature importance metrics for better model explainability.
- Scale efficiently for large datasets, particularly when using GPU acceleration.

4.3. Hyperparameter Tuning and Optimization

To ensure that our selected models achieved optimal performance, we employed nested cross-validation for hyperparameter tuning [2]. This two-level validation process minimizes bias and prevents overfitting:

1. **Outer loop:** Splits the dataset into training and validation folds to estimate the generalization error.
2. **Inner loop:** Uses `GridSearchCV` or `RandomizedSearchCV` to find the best hyperparameters.

This approach was used instead of traditional cross-validation to avoid data leakage and ensure that hyperparameter selection did not introduce bias into the performance evaluation.

5. Training and Evaluation

In this chapter, we detail the training process, model evaluation strategy, and final prediction generation. We emphasize the importance of properly structured training to ensure that the models generalize well to unseen data.

5.1. Dataset Splitting Strategy

To prevent data leakage and ensure a fair evaluation of model performance, we divided the dataset as follows:

- **Training Set (80% of data):** Used for model development and hyperparameter tuning.
- **Validation Set (20% of data):** Held out from training for final model evaluation.
- **External Test Set:** Used for generating final predictions.

This partitioning ensures that models are validated on unseen data before final deployment.

5.2. Nested Cross-Validation for Training

To improve the robustness of our models, we implemented **nested cross-validation**:

- **Outer loop (5 folds):** Divides the data into training and validation sets.
- **Inner loop (3 folds):** Performs hyperparameter tuning using `GridSearchCV` or `RandomizedSearchCV`.

This approach provides an unbiased estimate of model performance while identifying the best hyperparameters for training.

5.3. Final Evaluation on the Validation Set

After tuning hyperparameters, the best-performing models were evaluated on the validation set. Each model's RMSE was calculated:

```
with open('best_model_forest.pkl', 'rb') as f:
    best_model_rf = pickle.load(f)
y_val_pred_rf = best_model_rf.predict(X_val)
val_rmse_rf = sqrt(mean_squared_error(y_val, y_val_pred_rf))
print("Final Validation RMSE (Random Forest):", val_rmse_rf)
```

5.4. Generating Final Predictions

Once the models were validated, they were applied to the external test set to generate final predictions. The following procedure was used:

1. Drop non-relevant columns (e.g., CHASSIS_NUMBER).
2. Use the best-trained model to predict `LAID_UP_TIME`.
3. Save the results as an Excel file.

For example, the Random Forest model generated predictions as follows:

```
test_cleaned_copy = test_cleaned.copy()
chassis_number_rf = test_cleaned_copy['CHASSIS_NUMBER']
test_cleaned_copy = test_cleaned_copy.drop(columns=['CHASSIS_NUMBER', 'LAID_UP_TIME'])
y_test_pred_rf = best_model_rf.predict(test_cleaned_copy)
result_rf = pd.DataFrame({
    'CHASSIS_NUMBER': chassis_number_rf,
    'LAID_UP_TIME': y_test_pred_rf
})
result_rf.to_excel("ML/results/teamB-model2_RF.xlsx", index=False)
```

5.5. Comparison of Model Performance

A summary of model performance is presented in Table 2. The RMSE values indicate that XGBoost and Gradient Boosting (GPU variant) achieved the lowest error rates, making them the most effective models.

Model	cross-Validation RMSE	Train-Time [min]	Evaluation RMSE
Random Forest Regressor	37.69	24	37.65
Gradient Boosting (CPU)	34.24	220	34.56
Gradient Boosting (GPU)	34.22	170	34.42
Neural Network	NaN	NaN	51
Decision Tree	NaN	NaN	100

Table 2: Performance comparison of models based on RMSE

The ensemble models demonstrated superior performance over baseline models. Among them, Random Forest and GPU-accelerated Gradient Boosting provided the best predictive accuracy. The final

models were deployed for external test set prediction, ensuring a robust, data-driven approach to optimizing dealership efficiency.

6. Results and Discussion

The Gradient Boost algorithm was tested first, achieving a RMSE of 34 on the training dataset. However, the best performance was obtained with the RF algorithm, which yielded an RMSE of 73 on the test dataset.

A key factor influencing this result was the late-stage modifications applied to the RF algorithm. Due to time constraints, it was not possible to train the modified model using the entire dataset before the final submission. It is likely that a full training run would have resulted in a lower RMSE, further improving the model's predictive performance.

These findings highlight the importance of sufficient training time and computational resources when optimizing machine learning models.

7. Additional Questions

7.1. On which features/attributes does the laid up time in the car dealership depend? Which features have the biggest impact?

One of the key analytical questions investigated in this study was determining the features that influence the laid-up time in the car dealership and identifying those with the greatest impact. grupo... After performing multiple encoding tests, the target encoding was selected as the most suitable approach. Using its correlation matrix (Figure 9), it was observed that the attributes `VEHICLE_TYPE`, `UPHOLSTERY`, and `SCALED_INVENTURAL_VALUE` were the three most significant features.

Another method used to assess the importance of features was the built-in ranking provided by the Random Forest algorithm. According to this method (Figure 10), the top three influential features were `VEHICLE_TYPE`, `SCALED_INVENTURAL_VALUE`, and `PURCHASE_DATE_year`.

Both methods consistently ranked `VEHICLE_TYPE` as the most important feature. This consistency suggests that `VEHICLE_TYPE` has the greatest impact on the laid-up time in the car dealership.

7.2. What types of vehicles (based on their specifications) should be prioritized for purchase to minimize laid-up time at the dealership?

To understand the factors influencing the laid-up time of vehicles at a dealership and identify patterns that could help prioritize purchases to minimize this time. Initially, the target variable, `LAID_UP_TIME`, was a continuous numerical feature representing the number of days a vehicle remained at the dealership. To simplify the problem and make it more interpretable, we transformed this feature into a binary classification with two classes: "Short" and "Long", using the mean laid-up time as the threshold. Vehicles with laid-up times equal to or below the mean were classified as

"Short", while those above the mean were labeled as "Long". This approach allowed us to convert the problem into a classification task rather than a regression one.

Then a Decision Tree Classifier was trained to predict whether a vehicle would have a short or long laid-up time based on its features. To ensure that the tree remained interpretable and avoided overfitting, we set the maximum depth to 3, which limited the number of hierarchical splits in the decision process. The decision tree recursively split the data to find patterns that best separated the two classes, producing a set of decision rules.

To analyze the results, the decision tree was plotted (Figure 8) and examined the paths that led to predictions of "Short" laid-up times. From the tree, we derived several insightful rules:

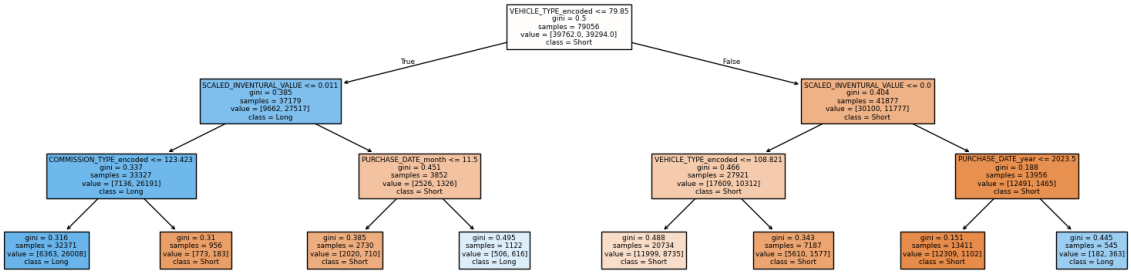

Figure 8: Resulting tree after classification.

- **Rule 1:** If `VEHICLE_TYPE_encoded` is greater than 79.85, `SCALED_INVETURAL_VALUE` is greater than 0, and `PURCHASE_DATE_year` is less than or equal to 2023.5, the `LAID_UP_TIME` is predicted to be smaller.
- **Rule 2:** If `VEHICLE_TYPE_encoded` is greater than 79.85, `SCALED_INVETURAL_VALUE` is less than or equal to 0, and `VEHICLE_TYPE_encoded` is either less than or equal to 108.821 or greater than 108.821, the `LAID_UP_TIME` is predicted to be smaller.
- **Rule 3:** If `VEHICLE_TYPE_encoded` is less than or equal to 79.85, `SCALED_INVETURAL_VALUE` is less than or equal to 0.011, and `COMMISION_TYPE_encoded` is greater than 123.423, the `LAID_UP_TIME` is predicted to be smaller.
- **Rule 4:** If `VEHICLE_TYPE_encoded` is less than or equal to 79.85, `SCALED_INVETURAL_VALUE` is greater than 0.011, and `PURCHASE_DATE_month` is less than or equal to 11.5, the `LAID_UP_TIME` is predicted to be smaller.

It is important to note that these rules are predictions based on patterns identified in the dataset and are not guaranteed outcomes. External factors not captured in the data might influence the laid-up time of vehicles. Additionally, the gini impurity values within the tree demonstrated how well the model separated the two classes, with lower gini values indicating better splits.

In conclusion, the decision tree model offered practical guidelines for vehicle purchase prioritization based on features such as vehicle type, purchase date, and inventory value. While these predictions are

valuable, they should be interpreted with caution, considering the inherent uncertainty in predictive models and the potential influence of external factors beyond the dataset.

8. Conclusion

This study aimed to develop a predictive model to estimate the *laid-up time* of vehicles at car dealerships based on historical sales data. By applying machine learning techniques, we sought to identify key factors influencing vehicle retention time and provide data-driven insights for optimizing dealership inventory.

8.1. Summary of Findings

Our analysis compared multiple machine learning models, including **Gradient Boosting** and **Random Forest**. Initially, the Gradient Boosting model achieved a training RMSE of 37.6. However, the best performance on the test dataset was obtained using Random Forest, which resulted in an RMSE of 73. This discrepancy can be attributed to the late-stage modifications applied to the RF model, which could not be fully trained before the final submission. Given additional computational resources and training time, further performance improvements would likely have been achieved.

Feature importance analysis revealed that `VEHICLE_TYPE`, `SCALED_INVENTURAL_VALUE`, and `PURCHASE_DATE_year` were the most influential attributes in predicting laid-up time. The consistency between correlation-based ranking and the feature importance ranking provided by Random Forest reinforces the significance of these features.

8.2. Implications for Dealerships

The findings of this study provide actionable insights for dealership management:

- **Inventory Optimization:** Dealerships should prioritize acquiring vehicle types that historically exhibit shorter laid-up times.
- **Pricing Strategies:** The relationship between `SCALED_INVENTURAL_VALUE` and *laid-up time* suggests that dynamic pricing models could enhance vehicle turnover.
- **Strategic Purchasing Decisions:** Vehicles purchased before specific time periods (e.g., `PURCHASE_DATE_year`) tend to have shorter storage durations, highlighting the importance of aligning inventory acquisition with demand trends.

8.3. Limitations and Future Research

Despite the promising results, several limitations must be acknowledged:

- **Incomplete Model Training:** Due to time constraints, the final RF model was not trained on the entire dataset, which likely impacted its performance.

- **Feature Engineering Enhancements:** Additional transformations, such as encoding optimizations and external data sources (e.g., market trends, economic indicators), could refine the predictive power.
- **Model Generalization:** While the model performed well on the given dataset, further validation is needed to assess its robustness across different dealership locations and market conditions.

8.4. Final Remarks

This study demonstrated the potential of machine learning in predicting vehicle laid-up time, providing valuable insights for dealerships to improve inventory turnover and pricing strategies. The results suggest that dealership operations can benefit from data-driven decision-making, ultimately enhancing efficiency and profitability.

Future work should focus on integrating external market data, improving model generalization, and exploring deep learning techniques for enhanced predictive accuracy.

A. Appendix

A.1. Feature understanding

- **RPAKREP_VEHICLE_HKEY**: Unique identifier for the vehicle record.
- **COMPANY**: The company associated with the vehicle.
- **OFFICE**: The office or branch managing the vehicle.
- **OFFICE_MAIN_BRAND**: The main brand represented by the office.
- **CHASSIS_NUMBER**: The chassis or VIN (Vehicle Identification Number) of the vehicle.
- **MANUFACTURER_SHORT**: Abbreviated name of the vehicle manufacturer.
- **MANUFACTURER**: Full name of the vehicle manufacturer.
- **VEHICLE_GROUP**: Category or classification of the vehicle.
- **VEHICLE_TYPE**: Specific type of vehicle (e.g., sedan, SUV, truck).
- **MODEL_CODE**: Internal or official model code of the vehicle.
- **VARIANT**: Specific version or trim level of the vehicle model.
- **MILEAGE**: The total distance traveled by the vehicle.
- **OPERATING_HOURS**: Total hours the vehicle has been in operation.
- **MILAGE_IN_FIELD**: Recorded mileage during field operations.
- **MILAGE_SALES**: Mileage recorded at the time of sale.
- **OPERATING_HOURS_SALES**: Operating hours recorded at the time of sale.
- **RIM_KEY**: Identifier or code for the vehicle's wheel rims.
- **COLOR_CODE**: Numeric or alphanumeric code representing the vehicle's color.
- **COLOR_CODE_NAME**: Name associated with the color code.
- **COLOR**: The general color description of the vehicle.
- **COLOR_TYPE**: Classification of the color (e.g., metallic, matte, pearl).
- **UPHOLSTERY_CODE**: Code representing the upholstery type or material.
- **UPHOLSTERY**: Description of the vehicle's upholstery (e.g., leather, fabric).
- **UPHOLSTERY_CODE_ALT**: Alternative code for the upholstery type.
- **CERTIFICATE_TYPE**: Type of certification or registration document for the vehicle.
- **CERTIFICATE_TYPE_DATE**: Date associated with the certification or registration.
- **FACTORY_NUMBER**: Factory-assigned identification number for the vehicle.

- **ENGINE_ID**: Unique identifier for the vehicle's engine.
- **ENGINE_TYPE**: Type of engine (e.g., gasoline, diesel, electric).
- **ENGINE_ID_ALT**: Alternative identifier for the engine.
- **TRANSMISSION**: Description of the vehicle's transmission system.
- **TRANSMISSION_TYPE**: Type of transmission (e.g., automatic, manual, CVT).
- **TRANSMISSION_ID**: Unique identifier for the transmission.
- **TRANSMISSION_SHORT**: Abbreviated transmission code or designation.
- **TRANSMISSION_NAME**: Full name or description of the transmission.
- **RIMS**: Type or specification of the vehicle's rims.
- **FRONT_TIRES**: Specifications of the front tires (e.g., size, brand).
- **FRONT_TIRES_CONDITION**: Condition or wear status of the front tires.
- **REAR_TIRES**: Specifications of the rear tires (e.g., size, brand).
- **REAR_TIRES_CONDITION**: Condition or wear status of the rear tires.
- **NUMBER_DOORS**: Total number of doors in the vehicle.
- **NUMBER_SEATS**: Total number of seats in the vehicle.
- **PERMITTED_TOTAL_WEIGHT**: Maximum legally permitted weight of the vehicle, including cargo and passengers.
- **MAX_TRAILOR_LOAD**: Maximum allowable trailer load the vehicle can tow.
- **CURB_WEIGHT**: Weight of the vehicle without passengers or cargo.
- **YEAR_CONSTRUCTION**: Year in which the vehicle was manufactured.
- **CONSTRUCTION_MONTH**: Month in which the vehicle was manufactured.
- **NUMBER_AXLE**: Total number of axles on the vehicle.
- **NUMBER_ENGINE_CYLINDER**: Number of cylinders in the vehicle's engine.
- **REPAIR_RKZ**: Indicator of whether the vehicle has undergone repairs.
- **OPTICAL_CONDITION**: Visual or exterior condition of the vehicle.
- **TECHNICAL_CONDITION**: Mechanical or operational condition of the vehicle.
- **ACCIDENT_VEHICLE**: Indicates whether the vehicle has been involved in an accident.
- **COMMISSION_NUMBER**: Internal commission number assigned to the vehicle.
- **HORSEPOWER**: Engine power measured in horsepower (HP).

- **KW**: Engine power measured in kilowatts (kW).
- **CCM**: Engine displacement in cubic centimeters (cc).
- **NUMBER_OWNERS**: Number of previous owners of the vehicle.
- **IS_USED_CAR**: Indicates whether the vehicle is used or new.
- **LEASING_CONTRACT_DATE**: Date when the leasing contract was signed.
- **LEASING_START**: Start date of the leasing period.
- **LEASING_END**: End date of the leasing period.
- **LEASING_MILAGE**: Mileage limit specified in the leasing contract.
- **PAINT_TYPE**: Type of paint used on the vehicle (e.g., metallic, matte).
- **FINANCING_TYPE**: Financing method or category for the vehicle.
- **FINANCING_TYPE_NAME**: Name or description of the financing type.
- **KAT_VEHICLE**: Vehicle classification or category code.
- **FUEL_TYPE**: Type of fuel the vehicle uses (e.g., gasoline, diesel, electric).
- **FUEL_TYPE_NAME**: Name or description of the fuel type.
- **DRIVE_TYPE**: Type of drivetrain (e.g., FWD, RWD, AWD).
- **DRIVE_TYPE_NAME**: Name or description of the drivetrain type.
- **VEHICLE_MODEL_ID**: Unique identifier for the vehicle model.
- **VEHICLE_MODEL_ID_NAME**: Name or description of the vehicle model ID.
- **COMMISSION_TYPE**: Type of commission related to the vehicle.
- **COMMISSION_TYPE_NAME**: Name or description of the commission type.
- **DEMONSTRATION_STATUS**: Indicates whether the vehicle has been used for demonstrations or test drives.
- **PURCHASE_DATE**: Date when the vehicle was purchased.
- **PURCHASE_BOOKING_DATE**: Date when the purchase was recorded in the system.
- **PURCHASE_MILAGE**: Mileage at the time of purchase.
- **PURCHASE_OPERATION_HOURS**: Operating hours at the time of purchase.
- **PRICE_LIST**: Price category or list to which the vehicle belongs.
- **DAY_OF_REGISTRATION**: Date when the vehicle was registered.
- **AT_LOCATION_SINCE**: Date since the vehicle has been at its current location.

- **LAID_UP_TIME**: Duration the vehicle has been inactive or not in use.
- **SOLD_CUSTOMER_ID**: Unique identifier of the customer who purchased the vehicle.
- **SOLD_INVOICE_CUSTOMER_ID**: Identifier of the customer associated with the sales invoice.
- **MILAGE_SALE**: Mileage recorded at the time of sale.
- **OPERATION_HOURS_SALE**: Operating hours recorded at the time of sale.
- **SOLD_INVOICE_CUSTOMER_ID2**: Alternative customer ID related to the sales invoice.
- **CUSTOMER_TYPE**: Classification of the customer (e.g., individual, corporate).
- **CUSTOMER_GROUP**: Group classification of the customer.
- **CUSTOMER_GROUP_NAME**: Name or description of the customer group.
- **CUSTOMER_FEATURE**: Specific attribute or characteristic of the customer.
- **CUSTOMER_FEATURE_NAME**: Name or description of the customer feature.
- **SALE_CUSTOMER_ID2**: Secondary identifier for the customer involved in the sale.
- **CUSTOMER_SALE_GROUP**: Group classification of the customer at the time of sale.
- **CUSTOMER_SALE_GROUP_NAME**: Name or description of the customer sale group.
- **CUSTOMER_SALE_GROUP2**: Alternative customer group classification at the time of sale.
- **CUSTOMER_SALE_GROUP2_NAME**: Name or description of the alternative customer sale group.
- **SCALED_CURRENT_VALUE**: Adjusted current value of the vehicle.
- **SCALED_INVENTURAL_VALUE**: Adjusted inventory value of the vehicle.
- **SCALED_REPORT_VALUE**: Adjusted reported value of the vehicle.
- **SCALED_VALUATION_PRICE**: Adjusted valuation price based on scaling factors.
- **SCALED_GUIDE_PRICE**: Adjusted guide price based on market or internal scaling.
- **SCALED_TOTAL_SALES_PRICE_BASIS**: Adjusted basis for the total sales price.
- **SCALED_TOTAL_SALE_PRICE**: Final adjusted total sales price of the vehicle.

A.2. Correlationmatrix and Feature Importance of RF

```
Correlation matrix with the target:
LAID_UP_TIME                    1.000000
VEHICLE_TYPE                    0.601461
UPHOLSTERY                      0.436305
SCALED_INVENTURAL_VALUE         0.430876
VEHICLE_GROUP                   0.328679
ENGINE_TYPE                     0.317253
TRANSMISSION_TYPE               0.313525
OFFICE                          0.269850
COMMISSION_TYPE                 0.265757
MANUFACTURER                    0.224528
OFFICE_MAIN_BRAND               0.182594
PURCHASE_BOOKING_DATE_month     0.144651
PURCHASE_DATE_month             0.142833
PURCHASE_DATE_year              0.132507
DRIVE_TYPE                      0.130382
PURCHASE_BOOKING_DATE_year      0.127280
AT_LOCATION_SINCE               0.115281
FUEL_TYPE                       0.106398
```

Figure 9: Correlation matrix with `LAID_UP_TIME` as the target.

	Feature	Importance
22	VEHICLE_TYPE_encoded	0.149286
13	SCALED_INVENTURAL_VALUE	0.133522
14	PURCHASE_DATE_year	0.068275
15	PURCHASE_DATE_month	0.059973
31	COMMISSION_TYPE_encoded	0.057588
24	UPHOLSTERY_encoded	0.056112
17	PURCHASE_BOOKING_DATE_month	0.049748
16	PURCHASE_BOOKING_DATE_year	0.041553
18	OFFICE_encoded	0.032716
25	ENGINE_TYPE_encoded	0.032106
21	VEHICLE_GROUP_encoded	0.031461
12	MILAGE_SALE	0.026347
11	AT_LOCATION_SINCE	0.026201
26	TRANSMISSION_TYPE_encoded	0.025305
1	MILEAGE	0.024379
2	MILAGE_SALES	0.023308
10	PURCHASE_MILAGE	0.022204
20	MANUFACTURER_encoded	0.017823

Figure 10: Feature importance according to the RF.

References

[1] Doe, John and Smith, Jane. 'Predictive Modeling in Automotive Industry: Applications and Insights'. In: *Journal of Machine Learning Applications* 15.3 (2023), pp. 45–62.

[2] Varma, Rakesh and Simon, Robert. 'Bias in error estimation when using cross-validation for model selection'. In: *BMC Bioinformatics* 7.1 (2006), p. 91. URL: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-7-91.