Mercedes-Benz Greener Manufacturing

https://www.kaggle.com/competitions/mercedes-benz-greener-manufacturing

PROBLEM FORMULATION

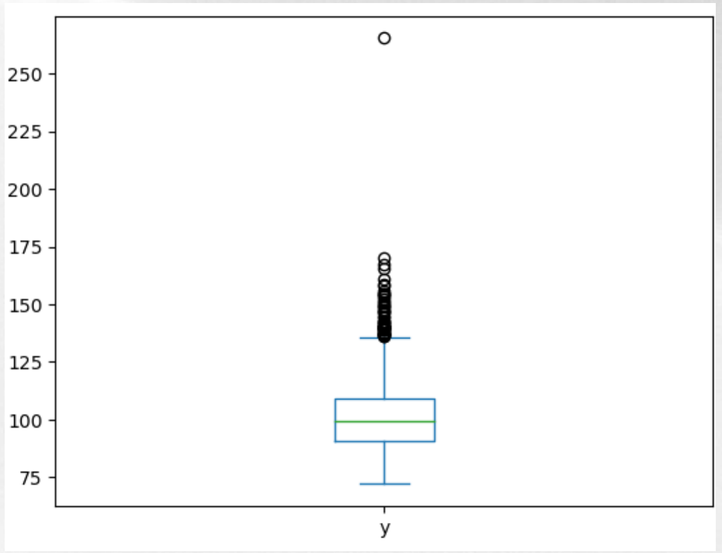
At the Mercedes factory, there is a facility dedicated to testing new vehicles. The testing time can vary considerably, ranging from 50 to 250 seconds. We have to predict this time using more than 300 different attributes. These attributes are anonymized, but their general meaning has been revealed by the organizers. They include eight categorical features that describe the characteristics of the car. The remaining attributes are mostly binary and refer to the characteristics of the tests conducted for each specific car.

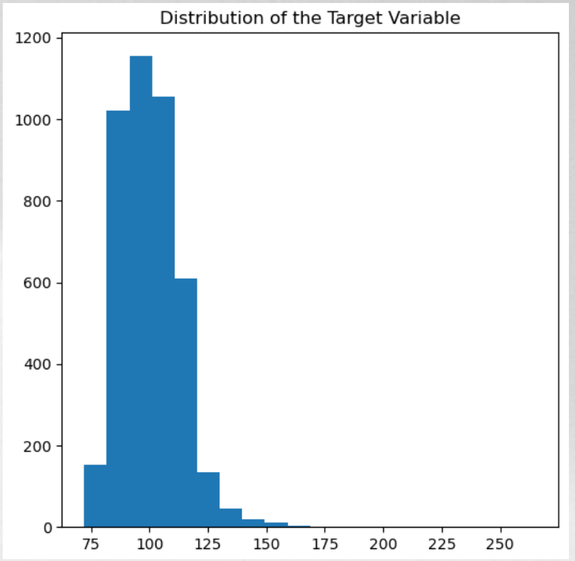
Analysis of the target variable and Data cleaning

The target variable has a standard distribution of about 72 to 140 seconds. The first and third quartiles lie between about 91 and 109 seconds, with a median of 100 seconds. Note that there are outliers starting at 140 seconds that we need to get rid of. Why it matters? Outliers can strongly influence the model, especially linear regression. They can skew the distribution of the data and lead to incorrect predictions.

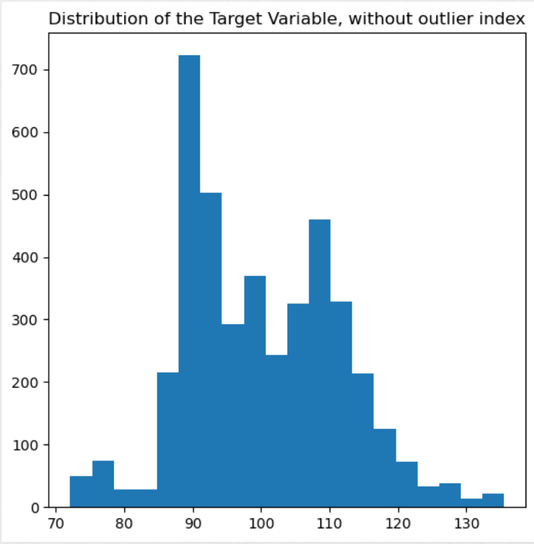
What we do: We calculate the first and third quartiles, then determine the lower and upper bounds based on the IQR. All values outside these boundaries are considered outliers and are removed from the dataset.

Before





After



Data preprocessing

Data Separation:
We split the data into training and validation samples in an 80/20 proportion.

Coding categorical features:
Many machine learning algorithms can't handle categorical features directly. Coding converts them into numeric values so that algorithms can use them.
What we do: We use OrdinalEncoder to encode categorical features. This method assigns unique numeric values to each category, which is appropriate for ordered categories.

Skip handling:
Missing values can lead to errors in models. Filling in omissions (imputation) allows all data to be used. We use SimpleImputer with a fill strategy with the most_frequent value (most_frequent) to fill gaps in the data.

Feature scaling:
Scaling brings all features to the same scale, which is especially important for models that are sensitive to feature scaling, such as linear regression.
We apply StandardScaler, which transforms the data so that it has a mean of 0 and a standard deviation of 1.

Selecting the best features:
Selecting the most informative features helps improve model performance by eliminating less meaningful features and reducing the dimensionality of the data.
We use SelectKBest with mutual_info_regression criteria to select the top 30 features.

Model training and evaluation

We trained a linear regression model on the trained data using LinearRegression.
And obtained the following results:

On train data = 0.6286849924753969
On validation = 0.6302262354298616