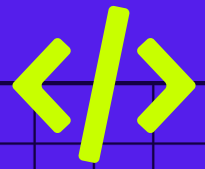
stackfuel

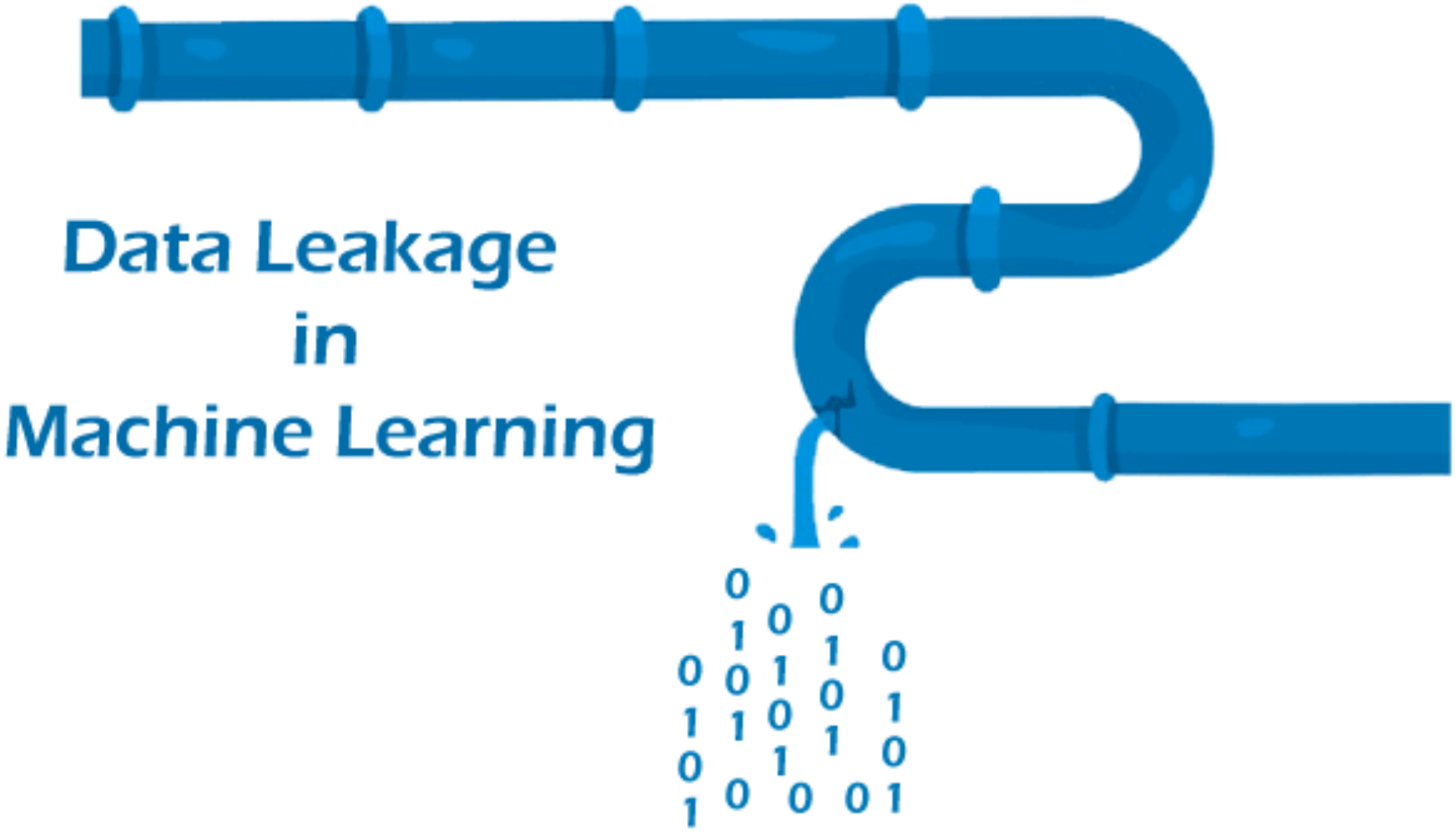
Data Leakage

Focus Session



stackfuel

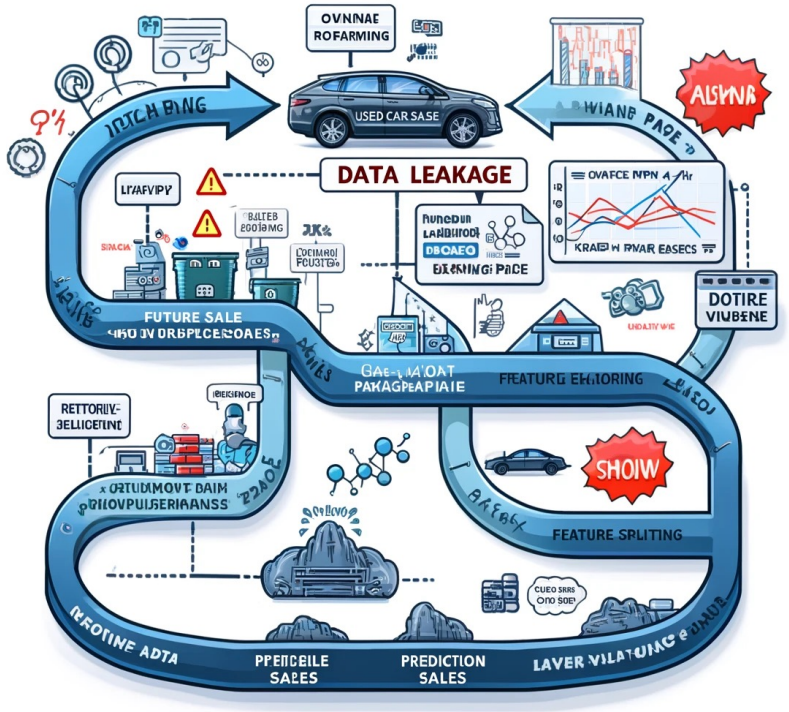
WHAT



https://prof-frenzel.medium.com/kb-data-cleaning-ii-de0e0ee630bb

Introduction

- Data leakage is a common pitfall in data science projects that can lead to overly optimistic model performance that doesn't generalize well to new data.

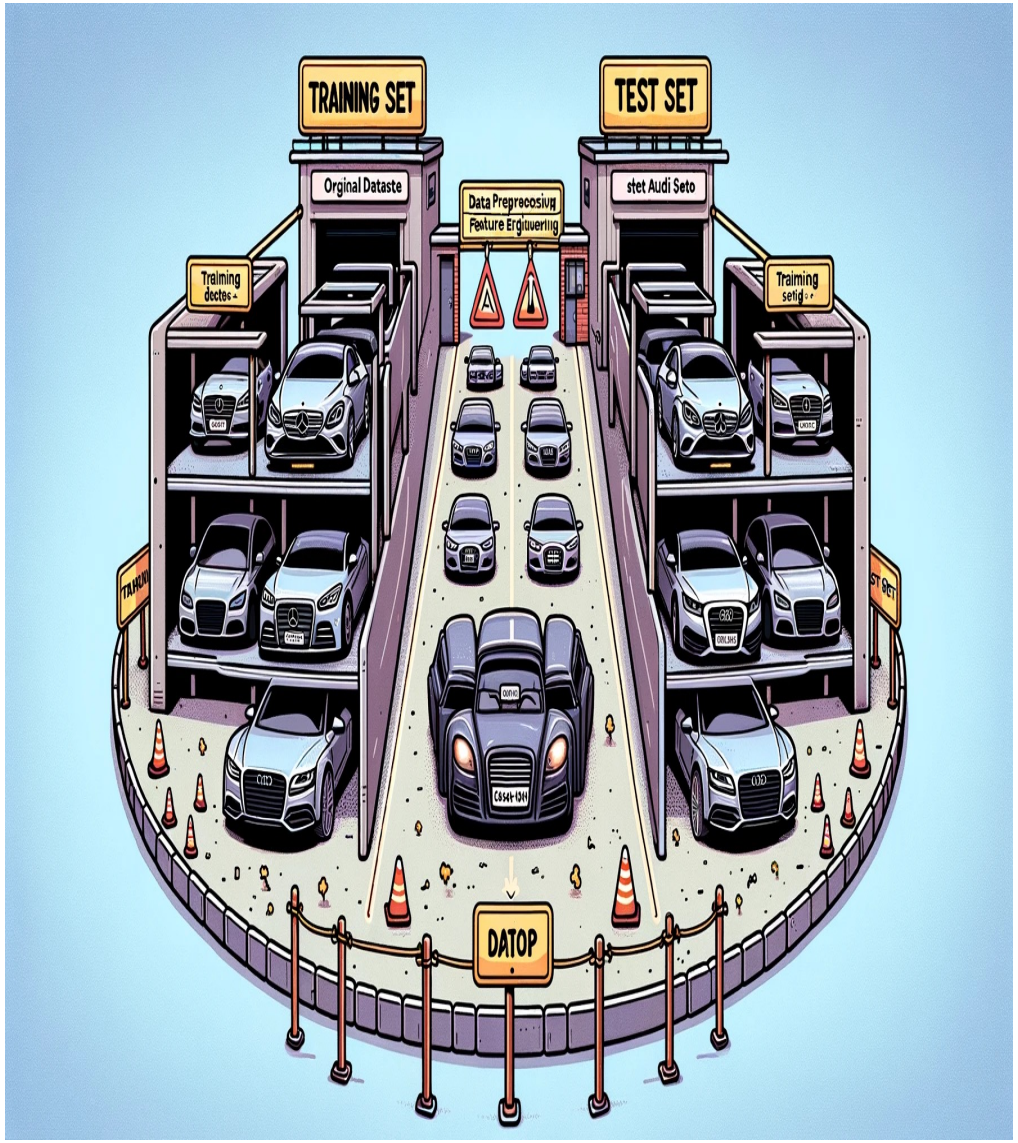


Splitting data incorrectly

stackfuel

Data Splitting:

- If you split your data into training and test sets after performing data pre-processing or feature engineering, you might inadvertently introduce information from the test set into your model.
- Always split your data into training and test sets before applying any transformations or creating new features. Use techniques like **train_test_split** from scikit-learn to ensure a proper split.

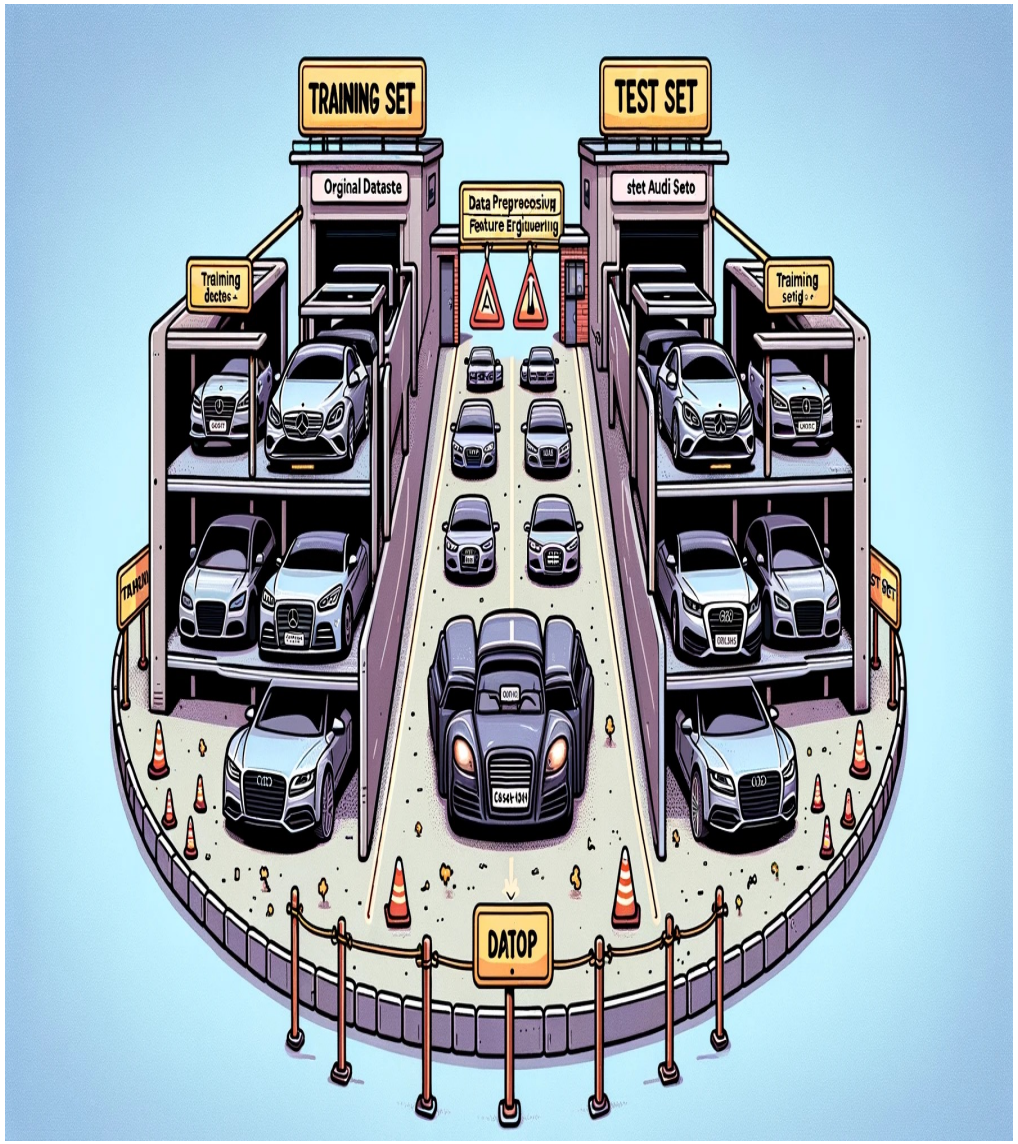


stackfuel

Pre-processing data before splitting

- If you apply pre-processing steps (e.g., scaling, normalization, encoding) to the entire dataset before splitting it into training and test sets, you're using information from the test set to make pre-processing decisions. Always split your data into training and test sets before applying any transformations or creating new features. Use techniques like **train_test_split** from scikit-learn to ensure a proper split.

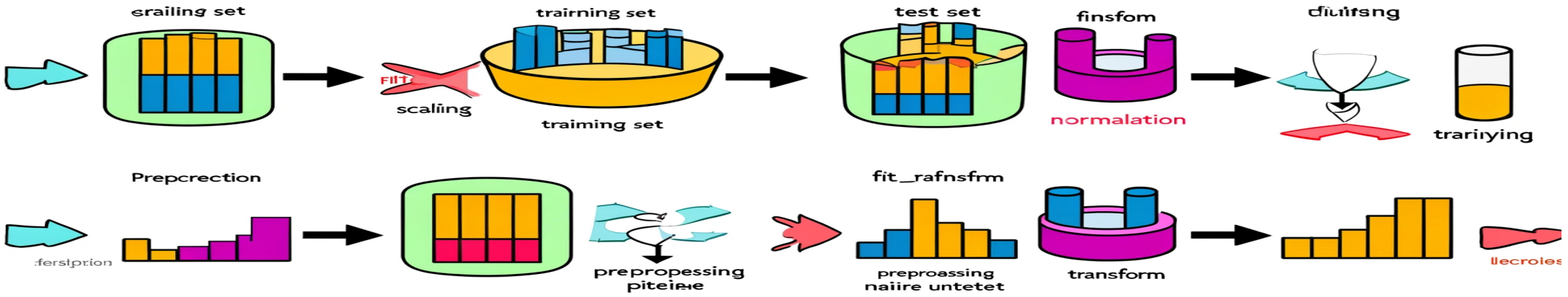
- Fit pre-processing steps only on the training data using methods like **fit transform**, and then apply the same transformations to the test data using **transform**. Use pipeline objects to encapsulate pre-processing steps and ensure consistent application.



stackfuel

Creating features using future information

- If you create features using information that would not be available at the time of prediction in a real-world scenario, you're introducing data leakage.
- Ensure that any features you create are based only on information that would be available at the time of prediction. Avoid using future data or data that would not be accessible in a production environment.



Leaking information through cross-validation

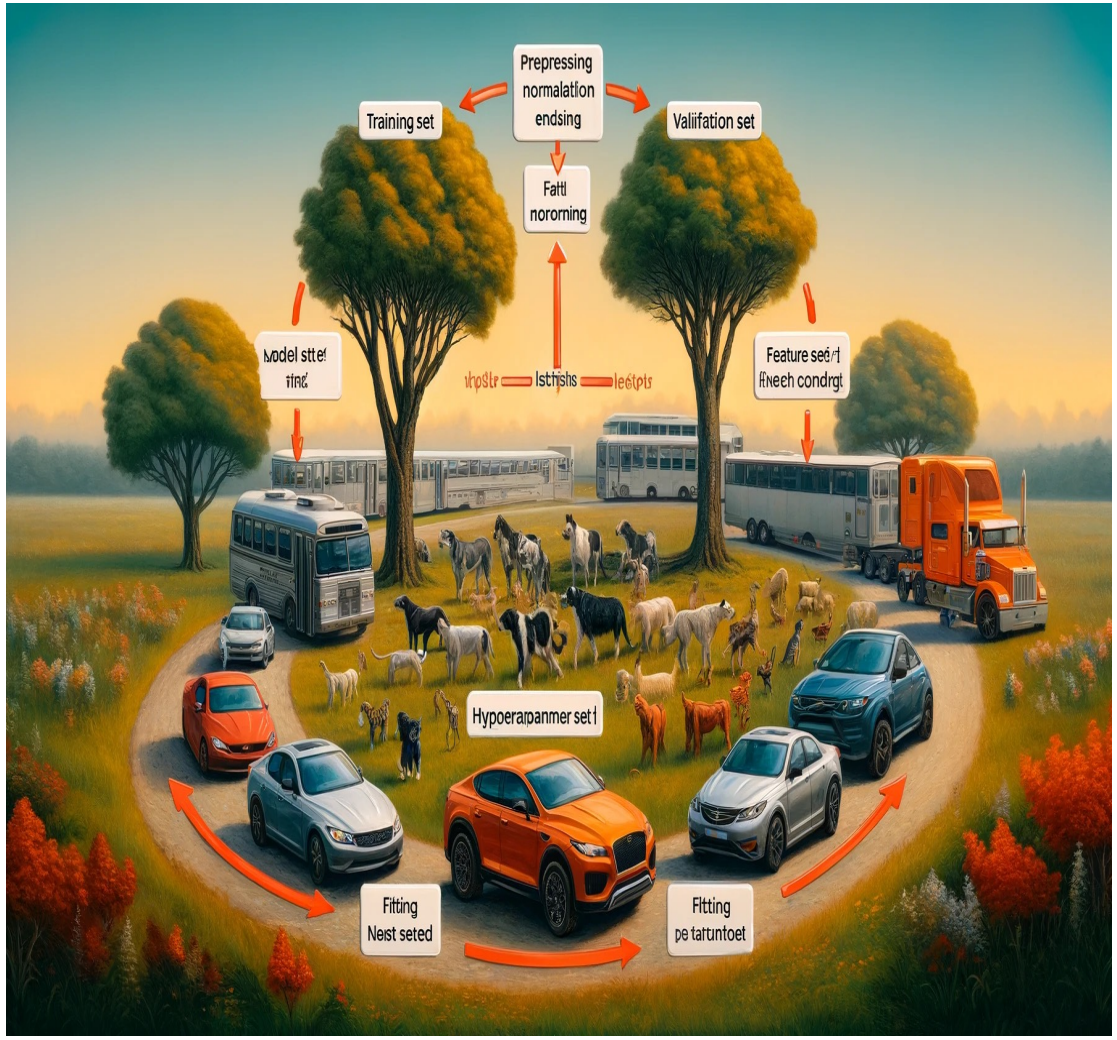
- If you perform data pre-processing or feature engineering outside the cross-validation loop, you might leak information between the training and validation folds.

- Apply pre-processing steps and feature engineering within each fold of the cross-validation process. Use pipeline objects to encapsulate these steps and ensure they are applied consistently within each fold.

Optimizing hyperparameters using the test set

- If you repeatedly evaluate your model on the test set and make hyperparameter tuning decisions based on the test set performance, you're indirectly fitting your model to the test set.

- Use a separate validation set or employ techniques like nested cross-validation for hyperparameter tuning. Make all model selection and hyperparameter tuning decisions based on the performance on the validation set, not the test set.



Leaking information through target variable

- If you create features that are directly derived from or highly correlated with the target variable, you're introducing data leakage..
- Avoid creating features that are a direct transformation or a proxy for the target variable. Ensure that the features are independent of the target variable and can be obtained without knowing the target value.

stackfuel

THANKS

Any Questions

https://link.springer.com/article/10.1007/s10694-020-00968-0