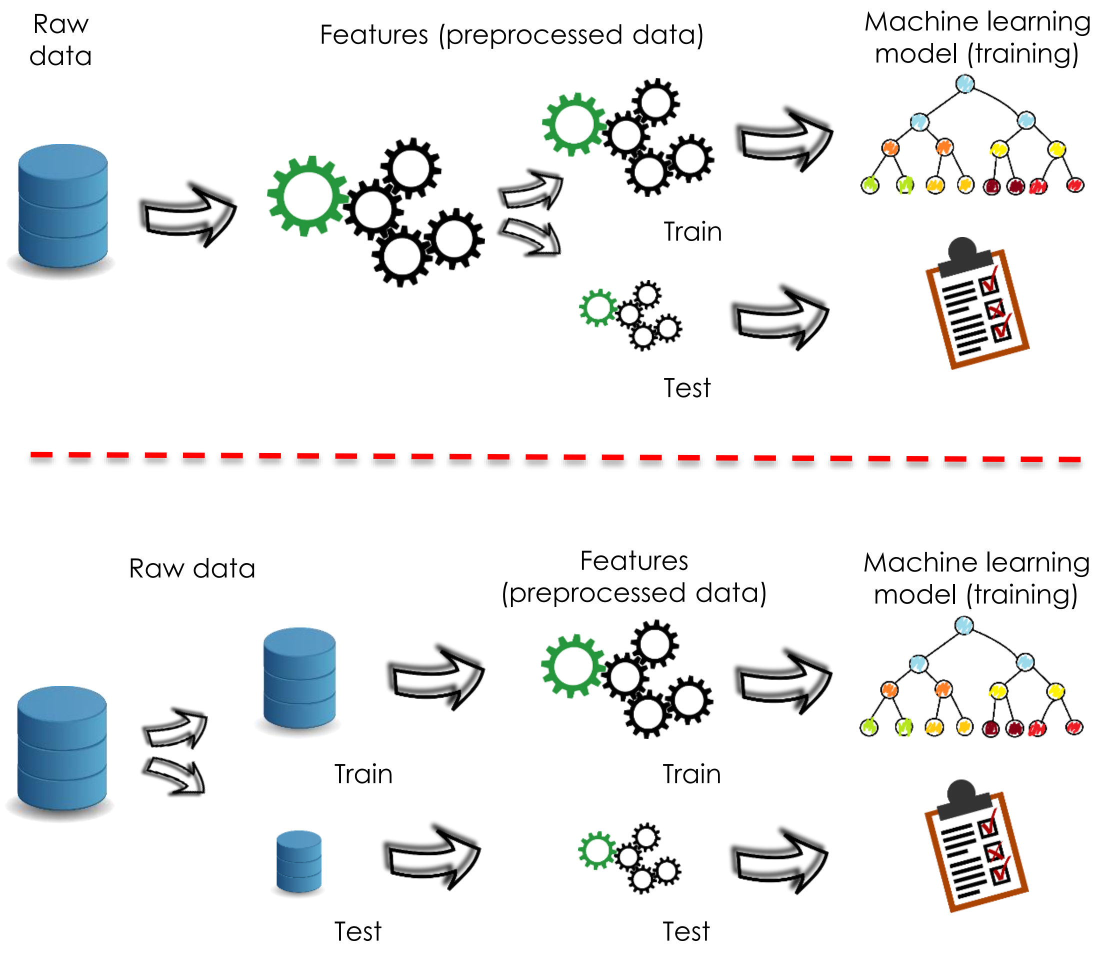
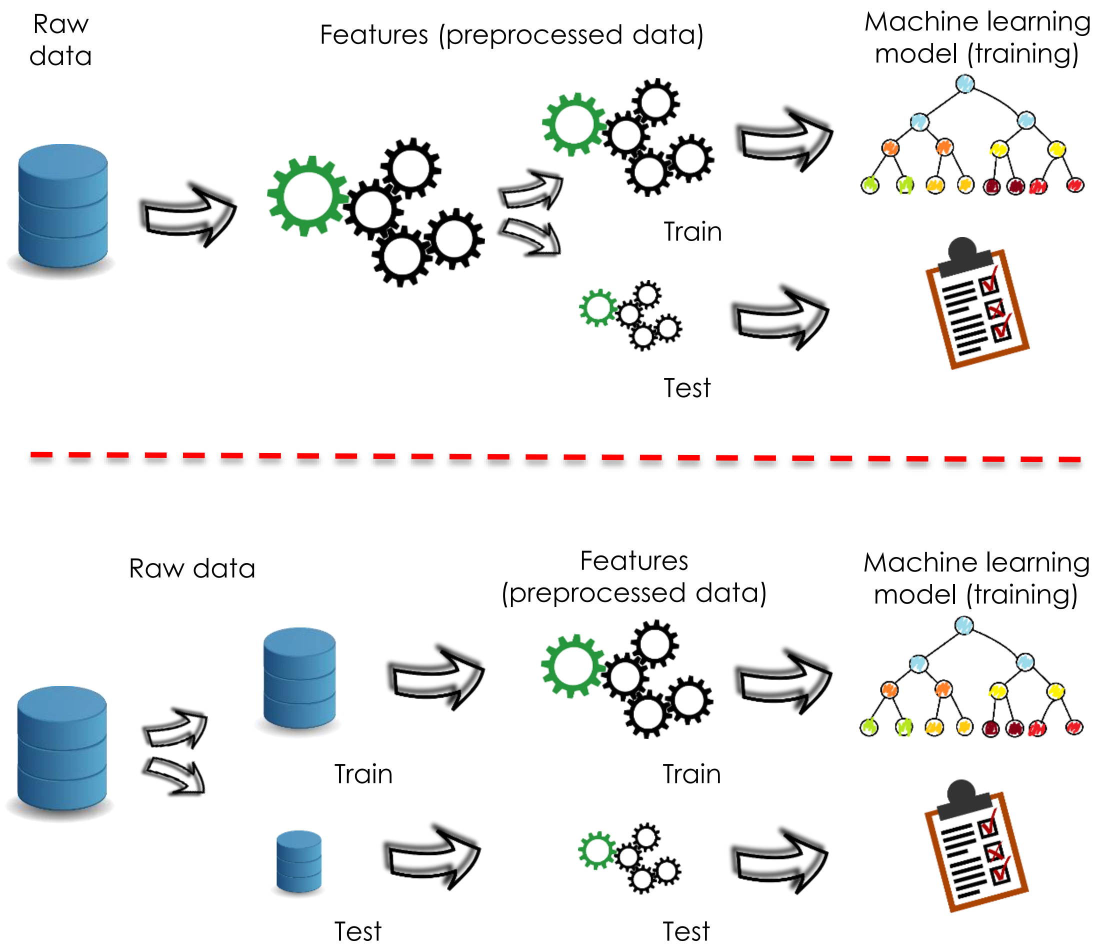
Feature Engineering

Tabular data

Machine learning workflow


Raw data
Features (preprocessed data)
Machine learning model (training)
Train
Test
Raw data
Features (preprocessed data)
Machine learning model (training)
Train
Train
Test
Test

Machine learning workflow: tabular data

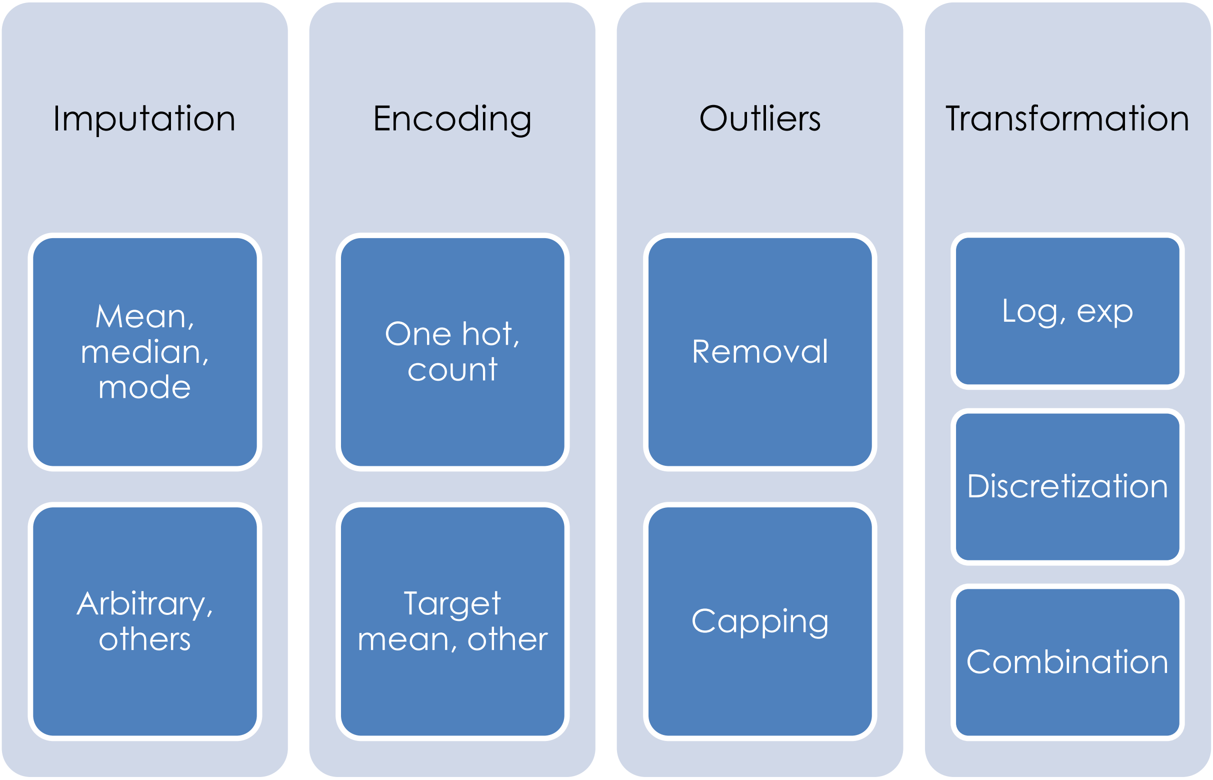

Raw data
Features (preprocessed data)
Machine learning model (training)
Train
Test
Raw data
Features (preprocessed data)
Machine learning model (training)
Train
Test
Train
Test

Tabular data

x1	x2	x3	x4	y

Each row is an independent observation.

Feature engineering – tabular data



Imputation

- Mean, median, mode
- Arbitrary, others

Encoding

- One hot, count
- Target mean, other

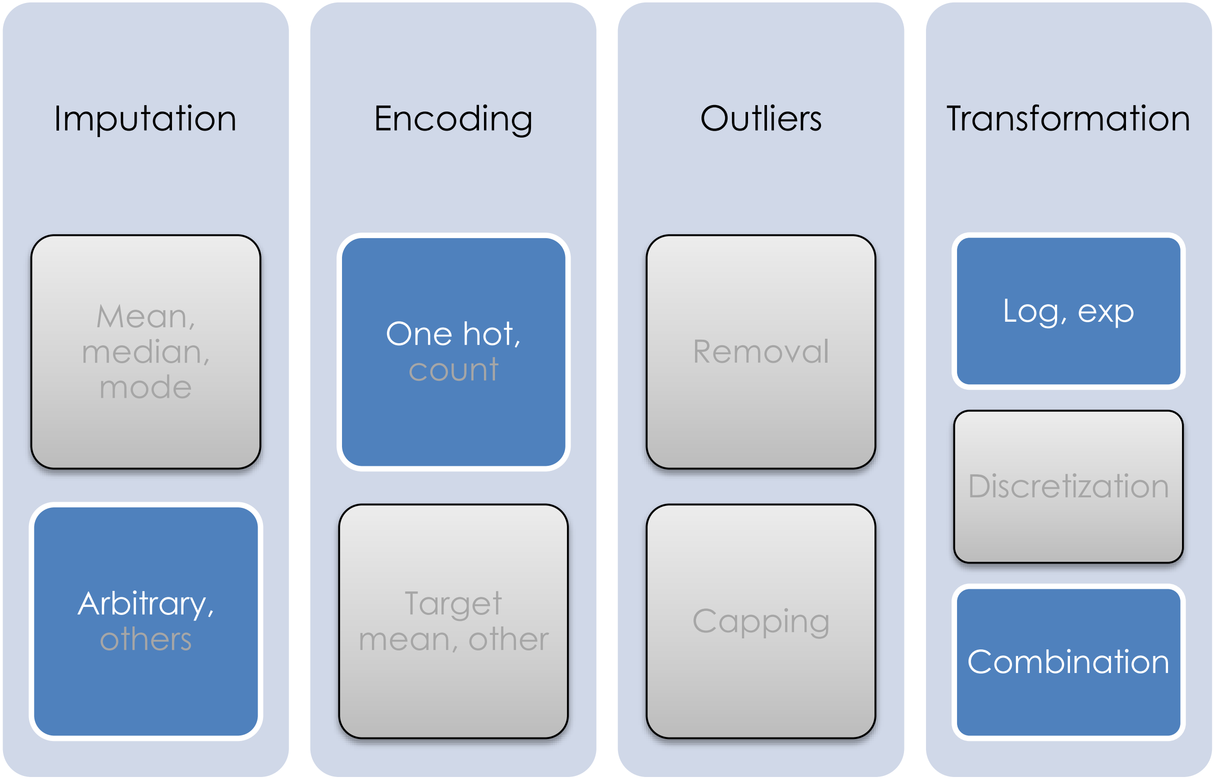
Outliers

- Removal
- Capping

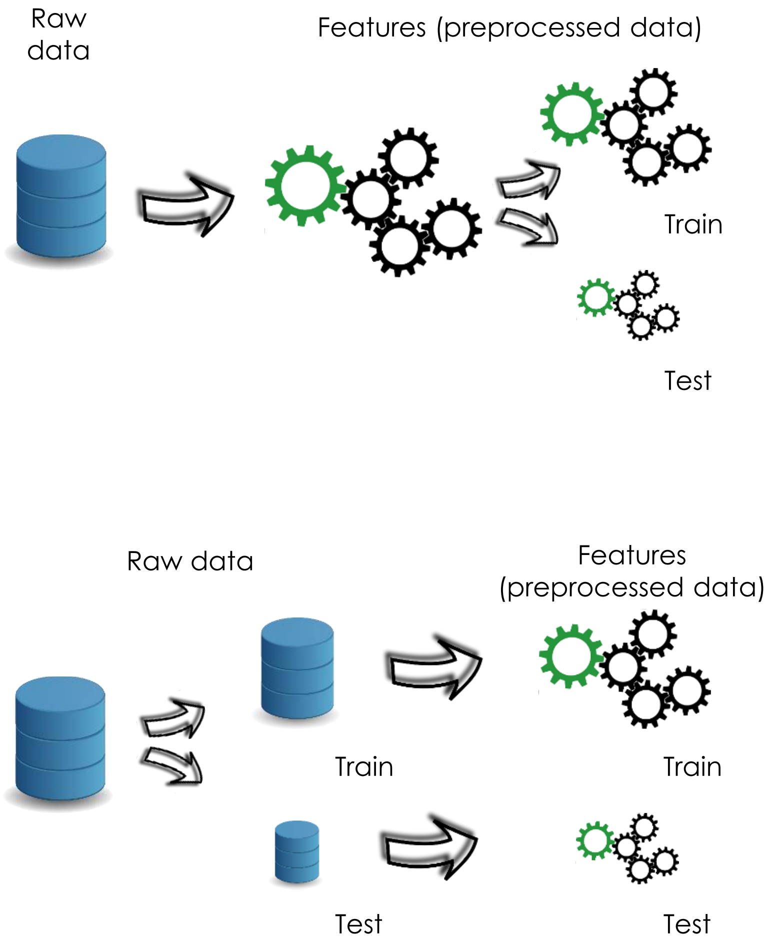
Transformation

- Log, exp
- Discretization
- Combination

Feature engineering – tabular data



Imputation	Encoding	Outliers	Transformation
Mean, median, mode	One hot, count	Removal	Log, exp
Arbitrary, others	Target mean, other	Capping	Discretization
			Combination

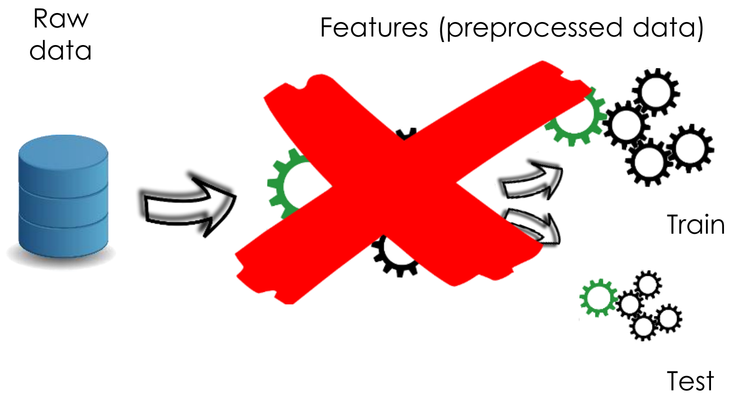


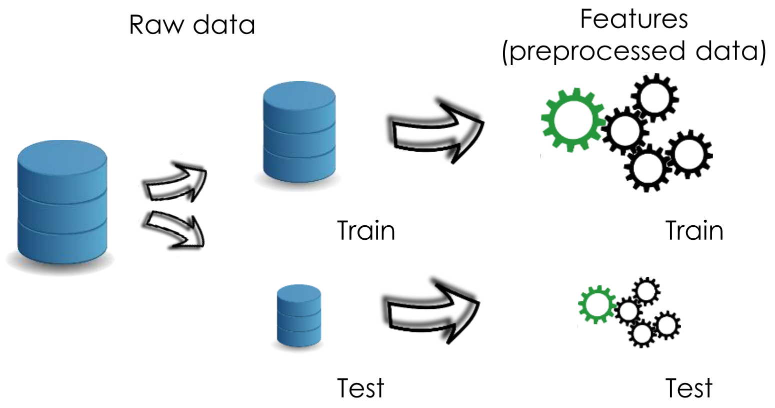
Some feature transformations **do not learn** parameters from data.

Doesn't matter when we split the data.

Feature engineering – tabular data

Imputation	Encoding	Outliers	Transformation
Mean, median, mode	One hot, count	Removal	Log, exp
Arbitrary, others	Target mean, other	Capping	Discretization
			Combination





Many feature transformations **learn** parameters from data.

Best to split the data before feature engineering.

Feature engineering libraries

scikit learn

Category Encoders

- fit() ➔ learns and stores parameters
- transform() ➔ transforms data

Feature engineering pipelines

scikit learn

Category Encoders

```
# set up pipeline
our_pipe = Pipeline([
            (“step1”, Imputation()),
            (”step2”, CategoricalEncoding()),
            (”step3”, Discretisation()),
            (”step4”, Scaling()),
            (”model”, Lasso()),
])

# train pipeline
Our_pipe.fit(X_train, y_train)

# predict
our_pipe.predict(X_train)
our_pipe.predict(X_test)
our_pipe.predict(any_data)
```

Feature engineering pipelines

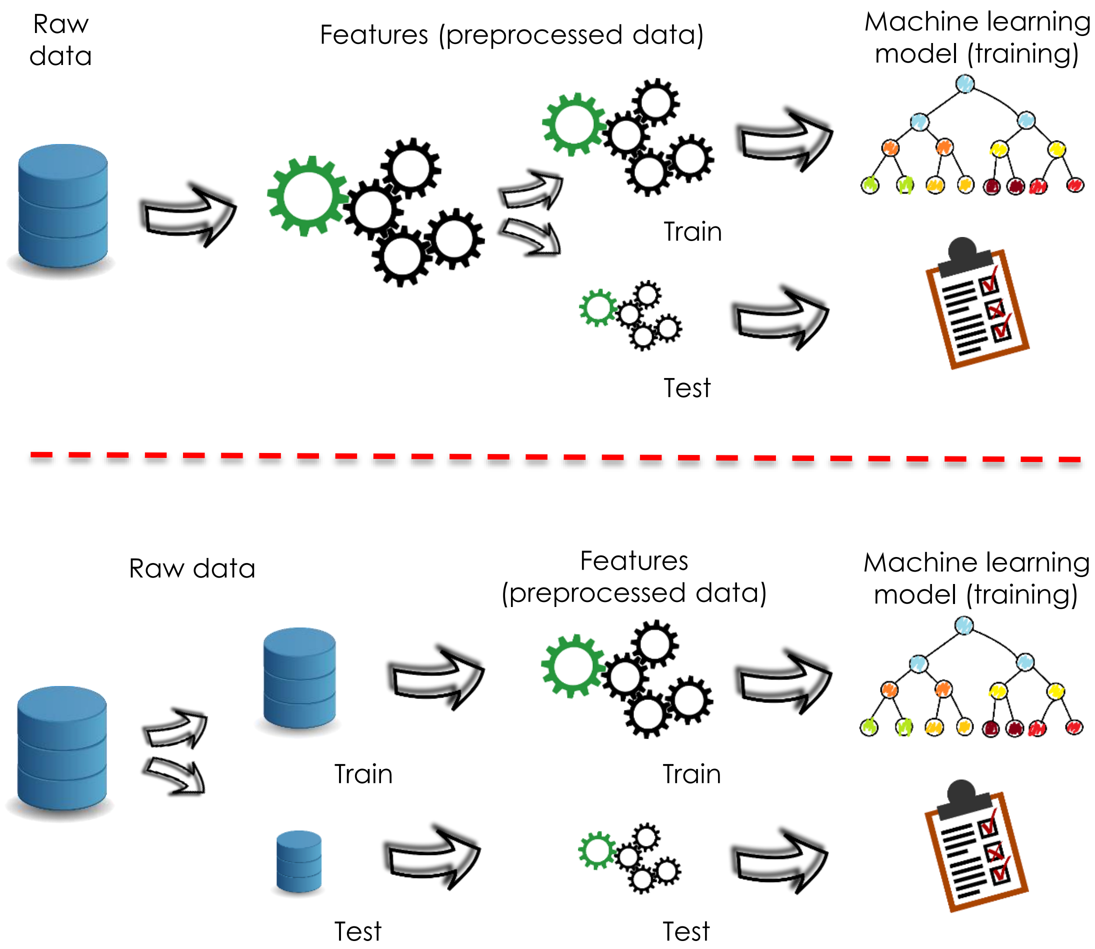
- We can pass raw data to the pipeline and obtain a prediction.
- We can deploy the pipeline to production.

```
# set up pipeline
our_pipe = Pipeline([
            (“step1”, Imputation()),
            (”step2”, CategoricalEncoding()),
            (”step3”, Discretisation()),
            (”step4”, Scaling()),
            (”model”, Lasso()),
])

# train pipeline
our_pipe.fit(X_train, y_train)

# predict
our_pipe.predict(X_train)
our_pipe.predict(X_test)
our_pipe.predict(any_data)
```

Machine learning workflow: forecasting


Raw data
Features (preprocessed data)
Machine learning model (training)
Train
Test
Raw data
Features (preprocessed data)
Machine learning model (training)
Train
Train
Test
Test

Time series data

y

The raw data does not contain the input features.

Summary

Many feature engineering procedures for tabular data learn parameters.

Best to split raw data before any transformation.

Feature transformation steps and machine learning model trained within a pipeline.

Pipeline can score any raw dataset and be deployed to production.