# Preprocessing and Pipelines

TM Quest

# Overview

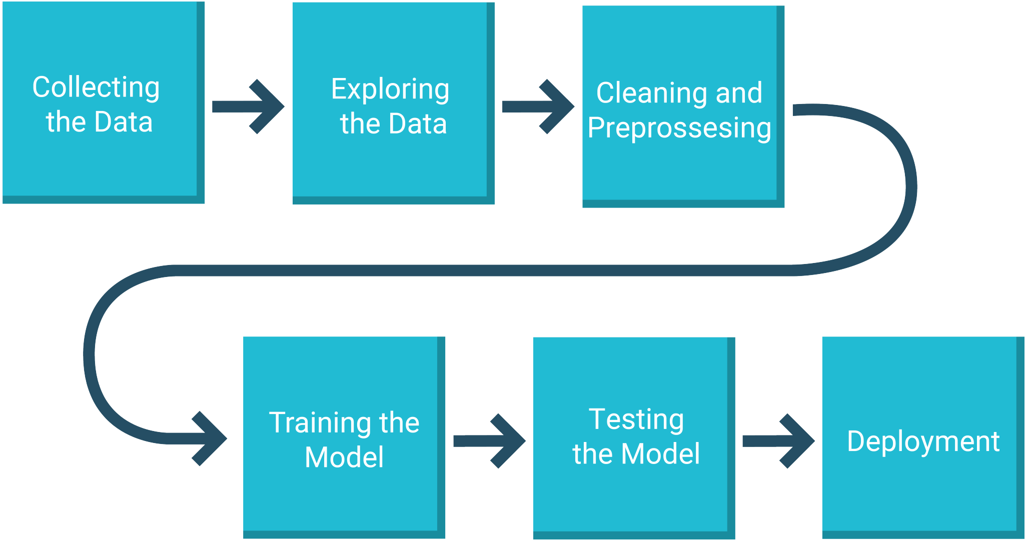
# Overview

## What Will we Learn in This Module?

- Preprocessing
  - Filling in missing data
  - Selecting useful features
  - Scaling features
- Pipelines
  - What is a pipeline?
  - How to create a pipeline?

# Preprocessing

# What is Preprocessing?



Collecting the Data

Exploring the Data

Cleaning and Preprossesing

Training the Model

Testing the Model

Deployment

# What is Preprocessing?

- Everything you do to make the data ready to be trained on.
- Example:
  - Dropping features
  - Combining features

## Filling in Missing Values

- Observations are often missing some features.
- For the training step the data need to be uniform.
- We can either drop the missing observations,
- or we can fill them in using for example the mean of each column.

# Missing Values

## Before

| Feature 1 | Feature 2 |
|---|---|
| 1 | NaN |
| NaN | 6 |
| 7 | 9 |
| 4 | 2 |
| 8 | 9 |
| NaN | 1 |
| 9 | 0 |
| 4 | NaN |

## After

| Feature 1 | Feature 2 |
|---|---|
| 1 | 4.7 |
| 5.5 | 6 |
| 7 | 9 |
| 4 | 2 |
| 8 | 9 |
| 5.5 | 1 |
| 9 | 0 |
| 4 | 4.7 |

# Standard Scaler

- Scales the data uniformly with mean 0 and variance 1.

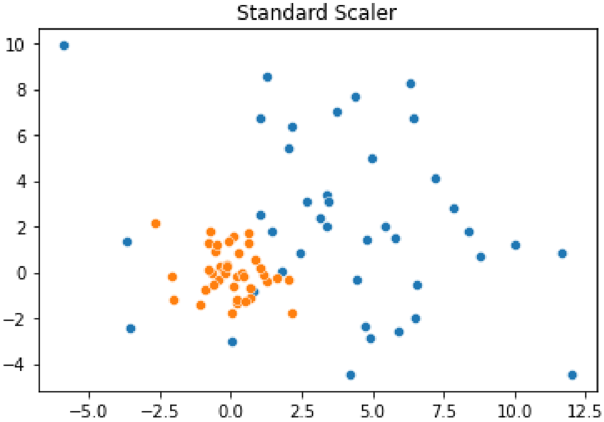


Figure: Blue: Before, Orange: after

# Code Examples

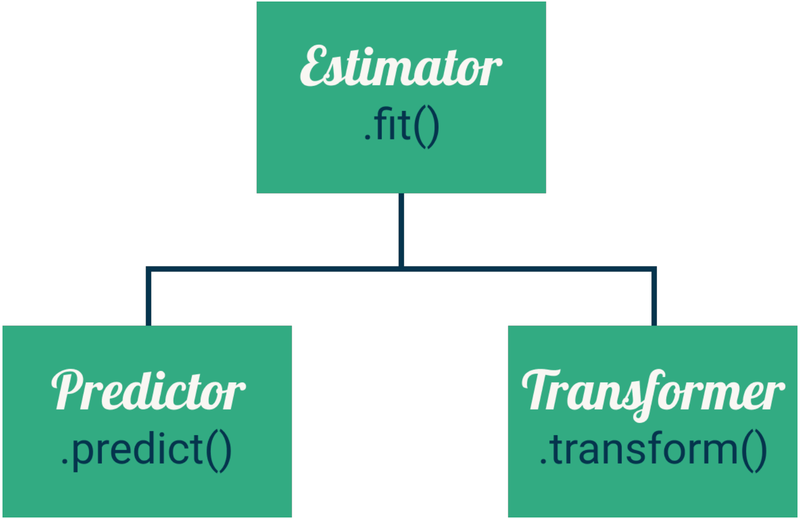
### Missing Values

```
# Drop observations with missing values
df["column"].dropna(inplace=True)

# Fill in missing values with the number 42
df["column"].fillna(value=42, inplace=True)
```

### Standard Scaler

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

# Scale the data
scaler.fit(data)
scaler.transform(data)
```

# Transformer



Estimator

.fit()

Predictor

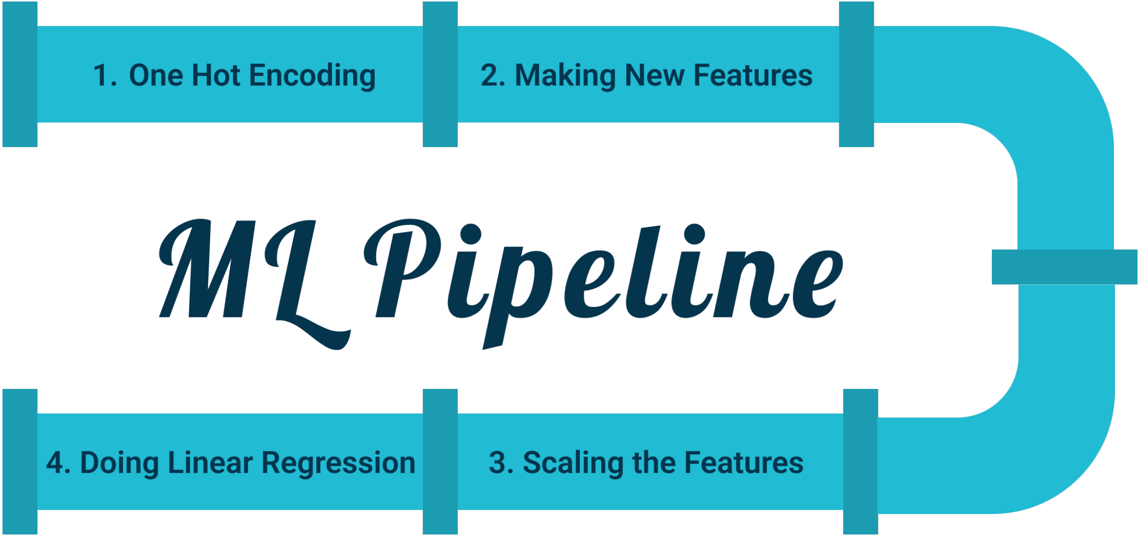
.predict()

Transformer

.transform()

# Pipelines

# ML Pipeline



1. One Hot Encoding

2. Making New Features

3. Scaling the Features

4. Doing Linear Regression

# Code Example

```python
from sklearn.pipeline import Pipeline

# Create a pipeline
pipeline = Pipeline([
    ("first_step", FirstTransformer()),
    ("second_step", SecondTransformer()),
    ("last_step", Predictor())])
```

The Pipeline now Works as a Predictor