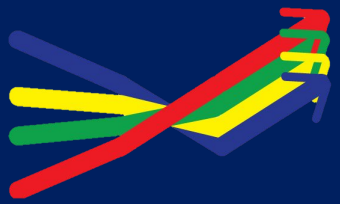
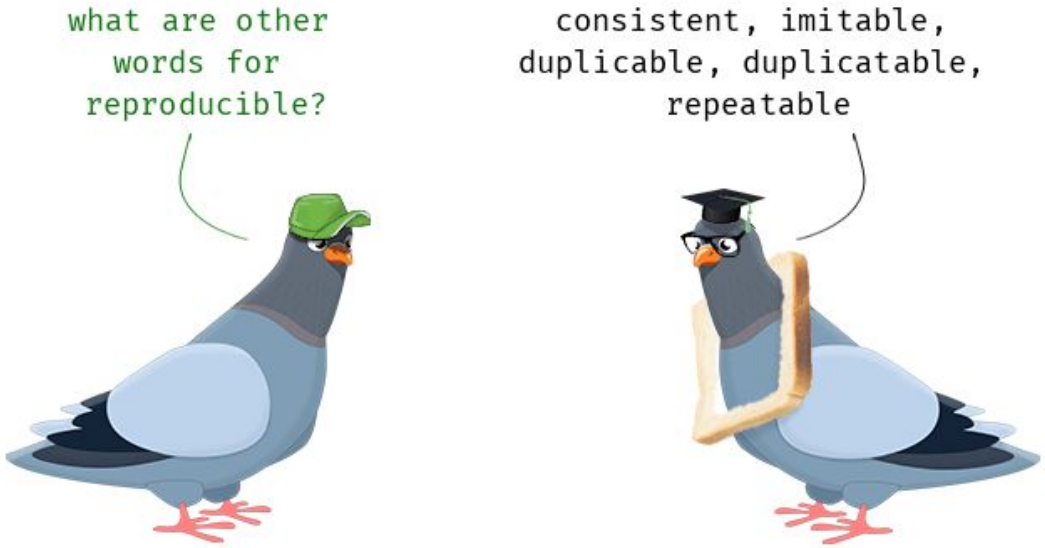
Building a Reproducible Machine Learning Pipeline

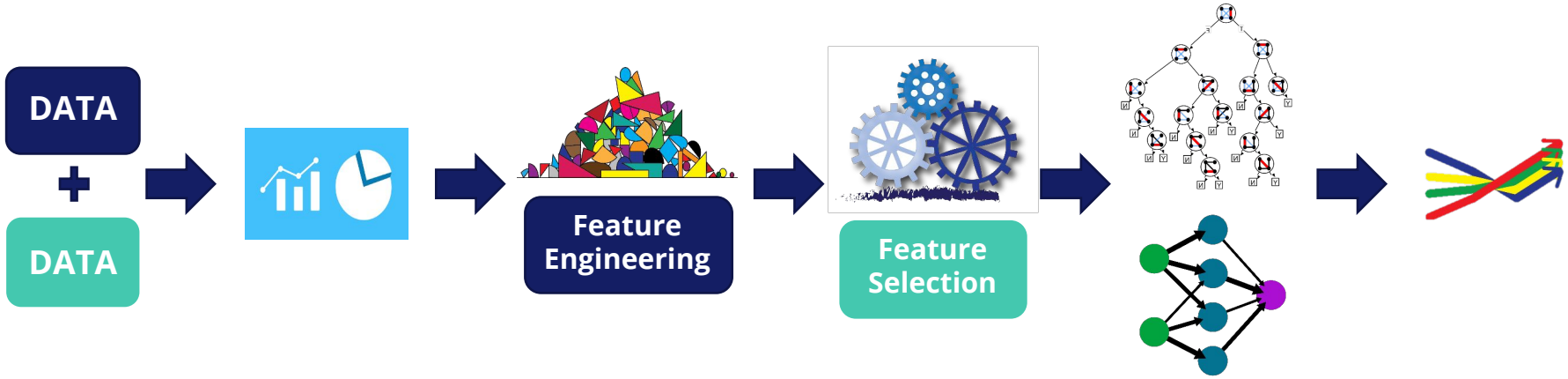


What is reproducibility in Machine Learning?



Reproducibility is the ability to duplicate a machine learning model exactly, such that given the same raw data as input, both models return the same output.

Machine Learning Pipeline: Overview



Gathering Data Sources

Data Analysis

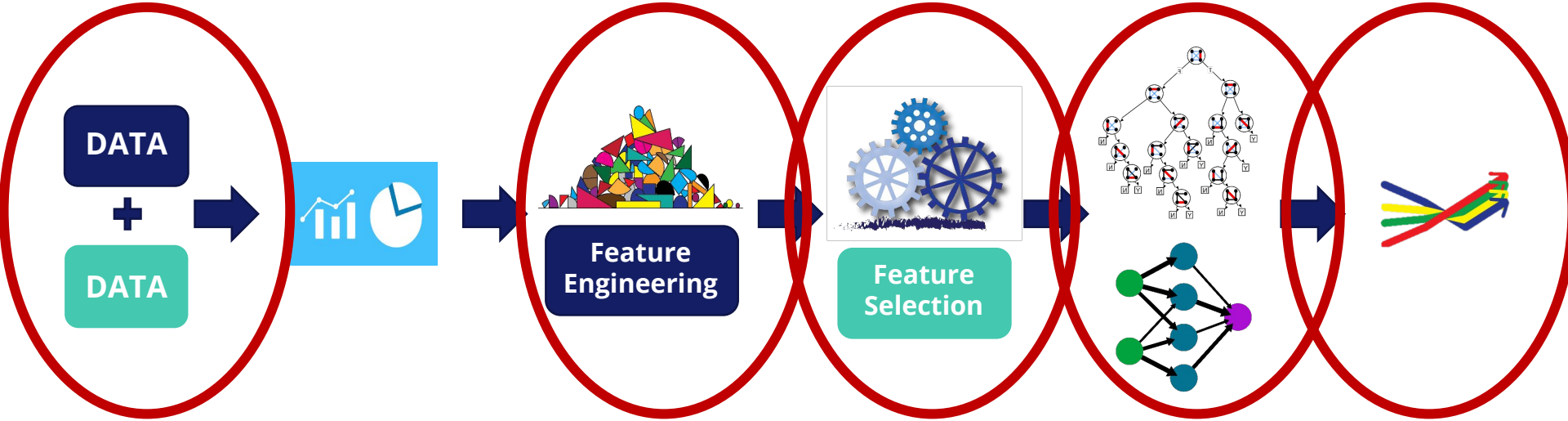
Data Pre-processing

Variable selection

Machine Learning Model building

Model deployment

Machine Learning Pipeline: Production



Gathering Data Sources

Data Analysis

Data Pre-processing

Variable selection

Machine Learning Model building

Model deployment

Reproducibility during data gathering

Gathering Data Sources

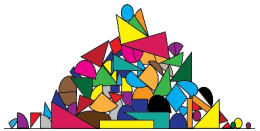
Data can be the most difficult challenge to ensure reproducibility

- ❖ Problems occur if the training dataset can't be reproduced at a later stage
- ➢ For example, the databases are constantly updated and overwritten, therefore values present at a certain time point differ from values later on.
- ➢ Order of data during data loading is random, for example when retrieving the rows with SQL.

How to ensure reproducibility?

- ❖ Save a snapshot of training data (either the actual data, or a reference to a storage location such as AWS S3
  - ✓ Good if the data is not pulled from too many different sources
  - ○ Potential conflict with GDPR
- ❖ Design data sources with accurate timestamps, so that a view of the data at any point in time can be retrieved
  - ✓ Ideal situation
  - ○ If not in house already, it requires a big effort to re-design the data sources

Reproducibility during feature creation



Feature Engineering

Data Pre-processing

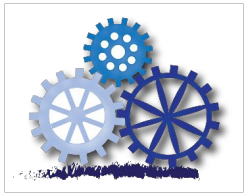
Lack of reproducibility may arise from:

- Replacing missing data with random extracted values
- Removing labels based on percentages of observations
- Calculating statistical values like the mean to use for missing value replacement
- More complex equations to extract features, e.g., aggregating over time

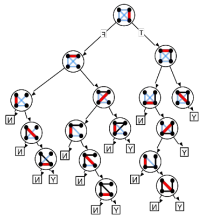
How to ensure reproducibility?

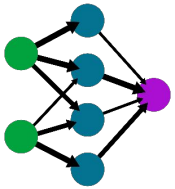
- Code on how a feature is generated should be tracked under version control and published with auto-incremented or timestamp hashed versions.
- Many of the parameters extracted for feature engineering depend on the data used for training ➔ ensure data is reproducible
- If replacing by extracting random samples, always set a seed

Reproducibility during model building



Feature Selection





How to ensure reproducibility?

- Record the order of the features
- Record applied feature transformations, e.g., standardisation
- Record hyperparameters
- For models that require an element of randomness to be trained (decision trees, neural networks, gradient descents), always set a seed.
- If the final model is a stack of models, record the structure of the ensemble.

Reproducibility during model deployment:
Software environment and implementation

Model deployment

How to ensure reproducibility?

- For full reproducibility, the software versions should match exactly - applications should list all third party library dependencies and their versions.
- Use a container and track its specifications, such as image version (which will include important information such as operating system version)
- Research, develop and deploy utilising the same language, e.g., python
- Prior to building the model, understand how the model will be integrated in production –how the model will be consumed-, so you can make sure the way it was designed can be fully integrated
  - Examples of partial deployment include, some data not being available at the time of consuming the model live
  - Filters in place do not allow a certain cohort of data to be seen by the model