Methodological topics Data-science specifics (part 2)

datalys

Ivan Srba

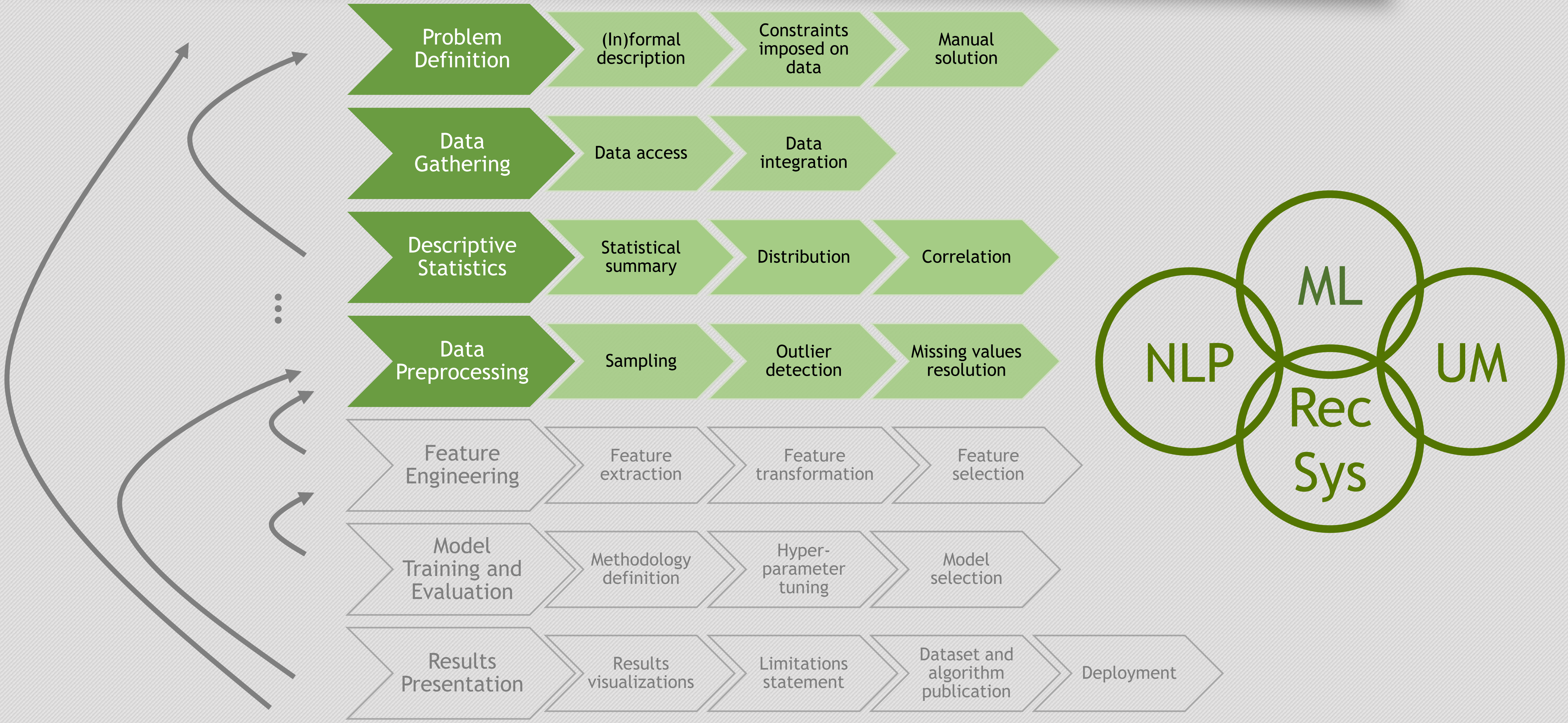
20th February 2019

PeWe@FIIT
personalized web group

Data-science specific questions…
(described in winter term)

- Data-science specific questions you need to answer before starting work on solution proposal and implementation:
  - How to define data-science (machine learning, …) task?
  - How to select/create appropriate dataset?
  - How to describe your dataset?
  - How to preprocess your dataset?
  - …

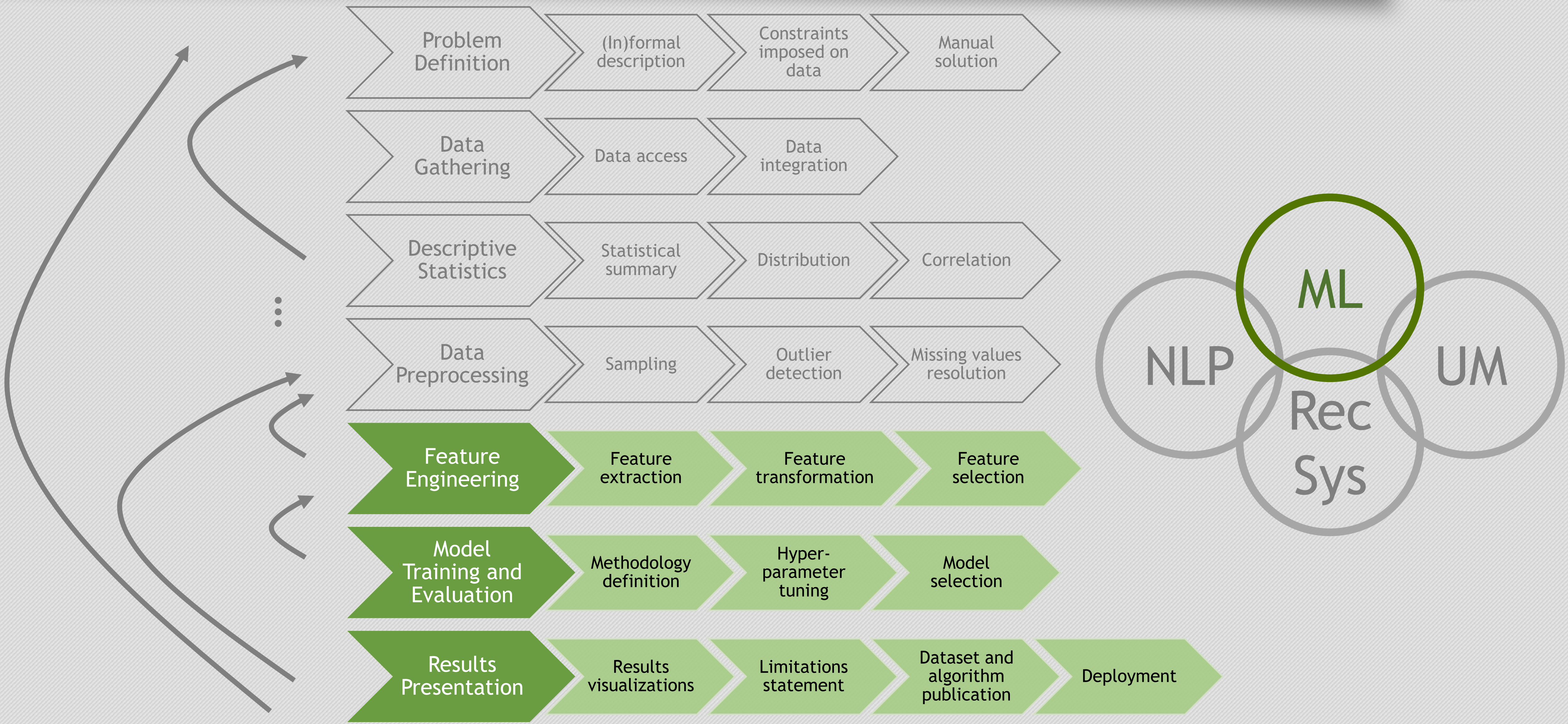
ML workflow: Generic part (described in winter term)



Warming-up

- Everything said last week applies perfectly also in case of all theses in data science domain
- Summary of gold rules
  - Explicitly state your goals
  - Describe your proposal conceptually, an in more details afterwards
  - Split method proposal from its implementation and evaluation
  - Define experimental methodology (evaluation steps, metrics, etc.)
  - Select appropriate baseline
  - Discuss results
  - Explicitly state possible limitations of your method
  - Pay attention to conclusions and appendixes

ML workflow: ML-specific part



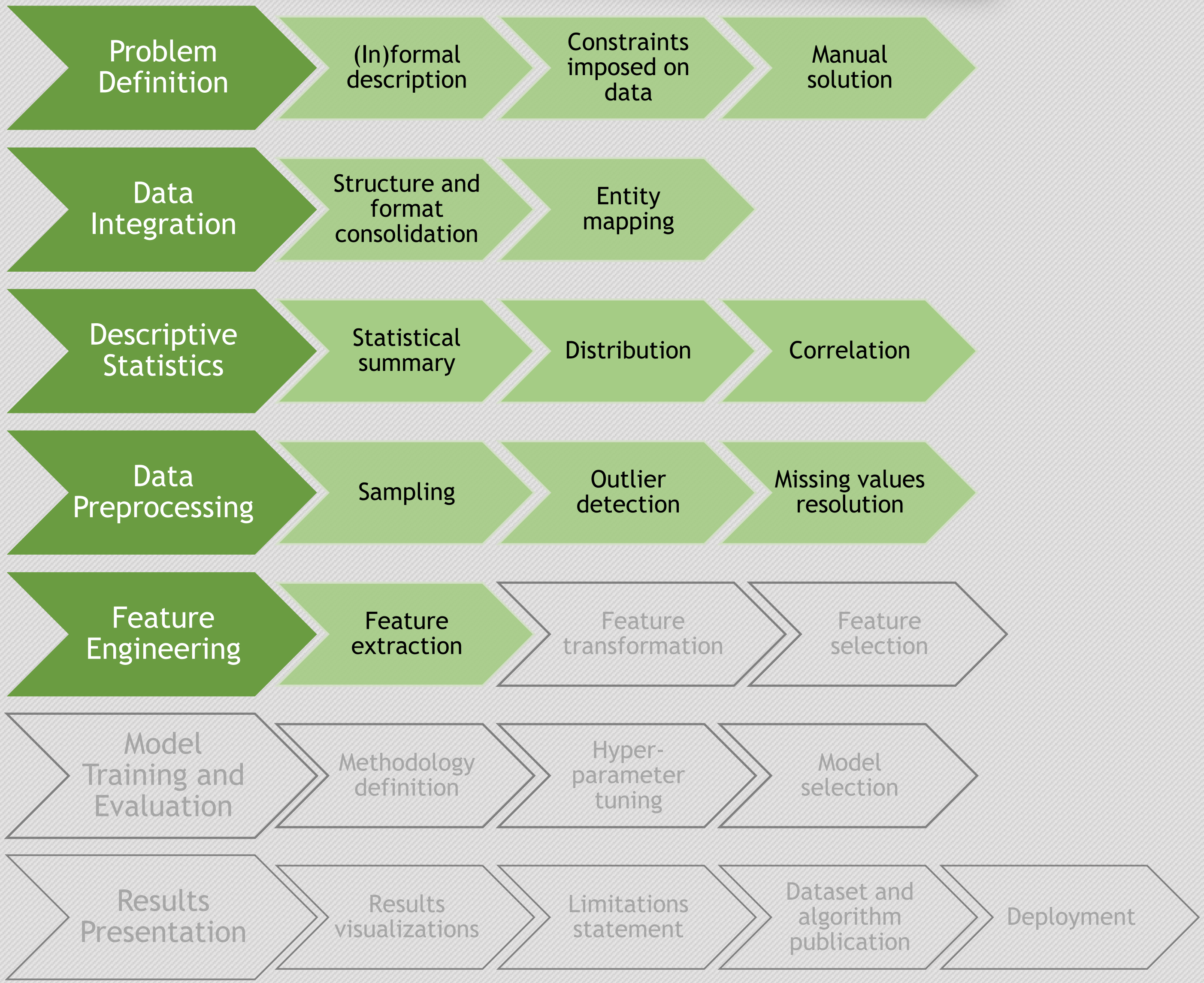
"Coming up with features is difficult, time-consuming, requires expert knowledge. *Applied machine learning* is basically feature engineering."

Andrew Ng

Feature Engineering

Feature extraction

- Raw and high-dimensional data (images, text, logs, …) need to be reduced and converted to features
- Techniques
  - Expert-based (UM, NLP)
  - Dimensionality-reduction (PCA)
  - Automatized
  - …
- Hints
  - You cannot skip this step, but it can be done iteratively and incrementally

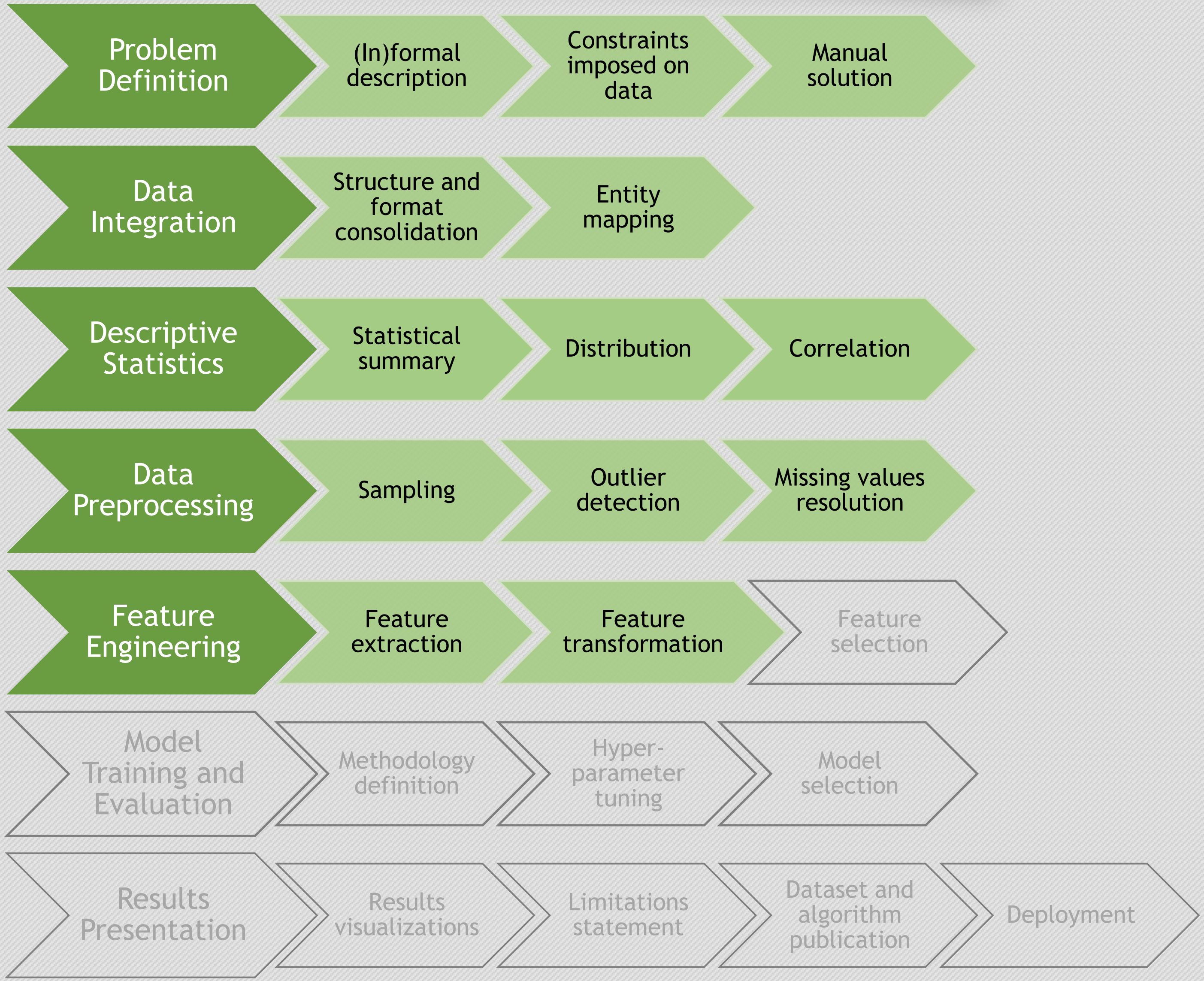


Feature transformation

- Features must have specific distribution, range or data type to work well with some ML algorithms

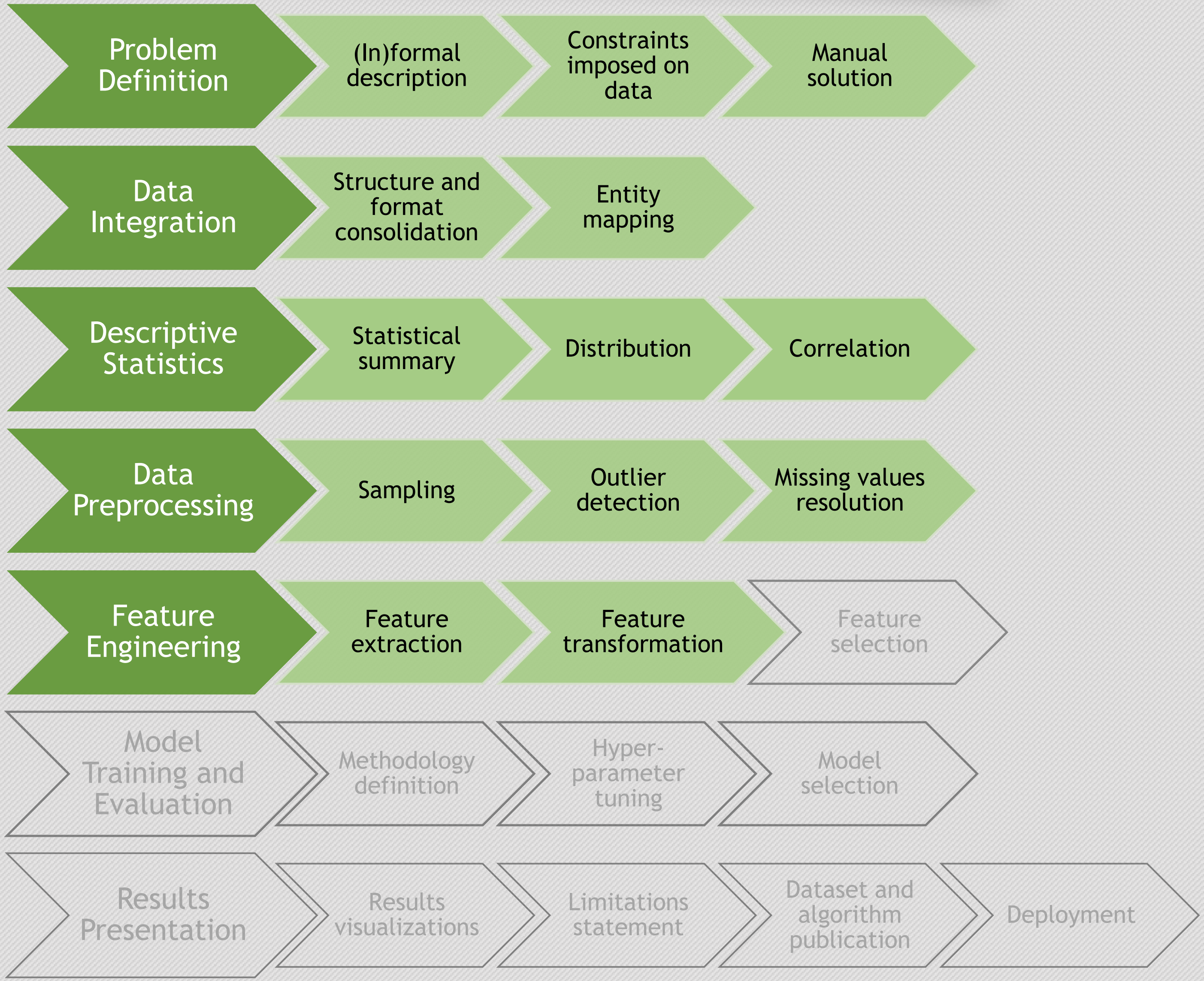
- Techniques
    - Scaling
    - Normalization
    - Binarization of features
    - Splitting features (e.g. date)
    - Encoding categorical features

- Hints
    - Start with ML algorithms which have less requirements on data distribution, range or data type (e.g. decision trees, random forest)



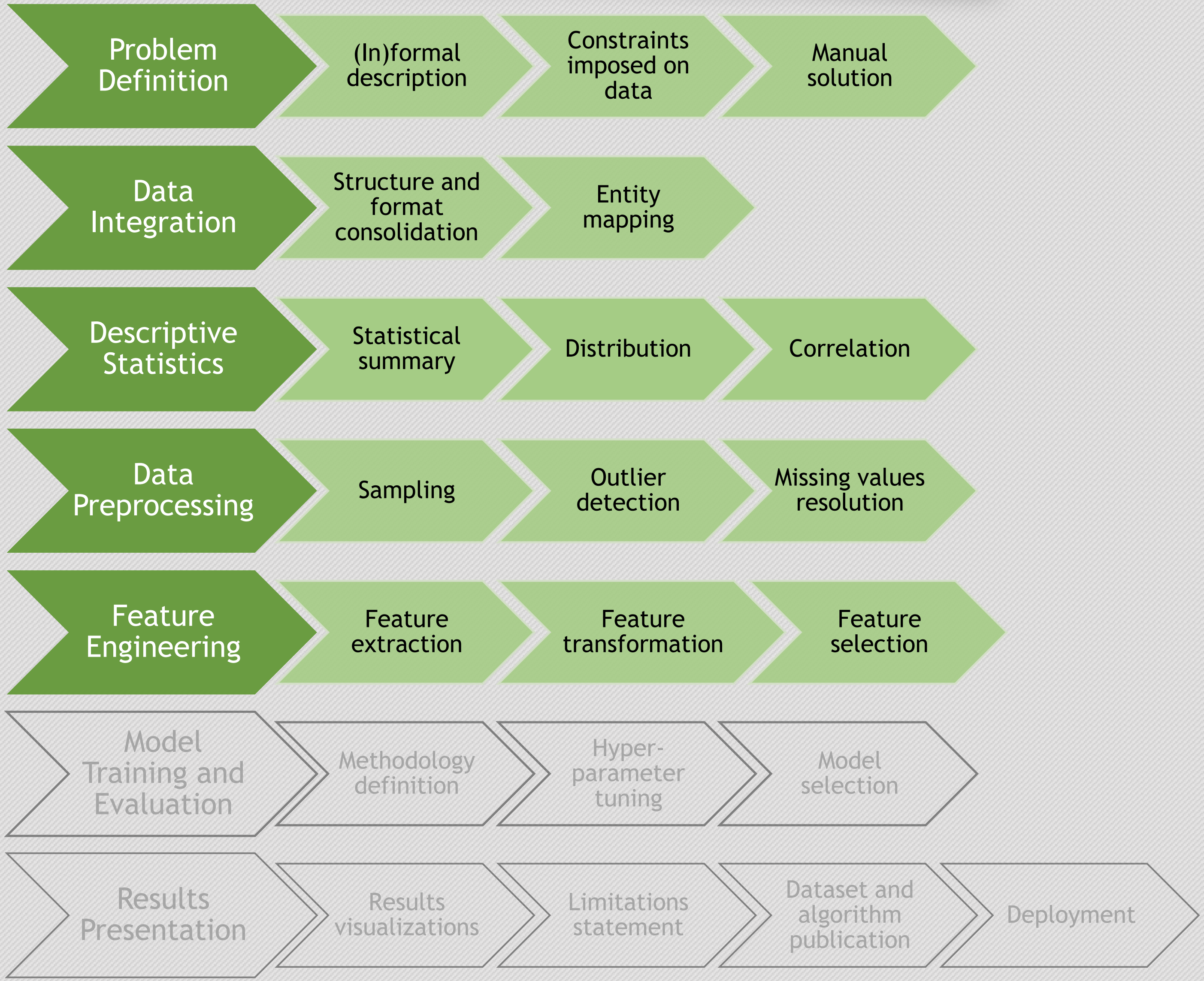
Feature transformation

- New useful features can be created from combination of existing features
- Techniques
  - Combining features
  - Polynomial features
- Hints
  - In the first iterations, completely skip this step



Feature selection

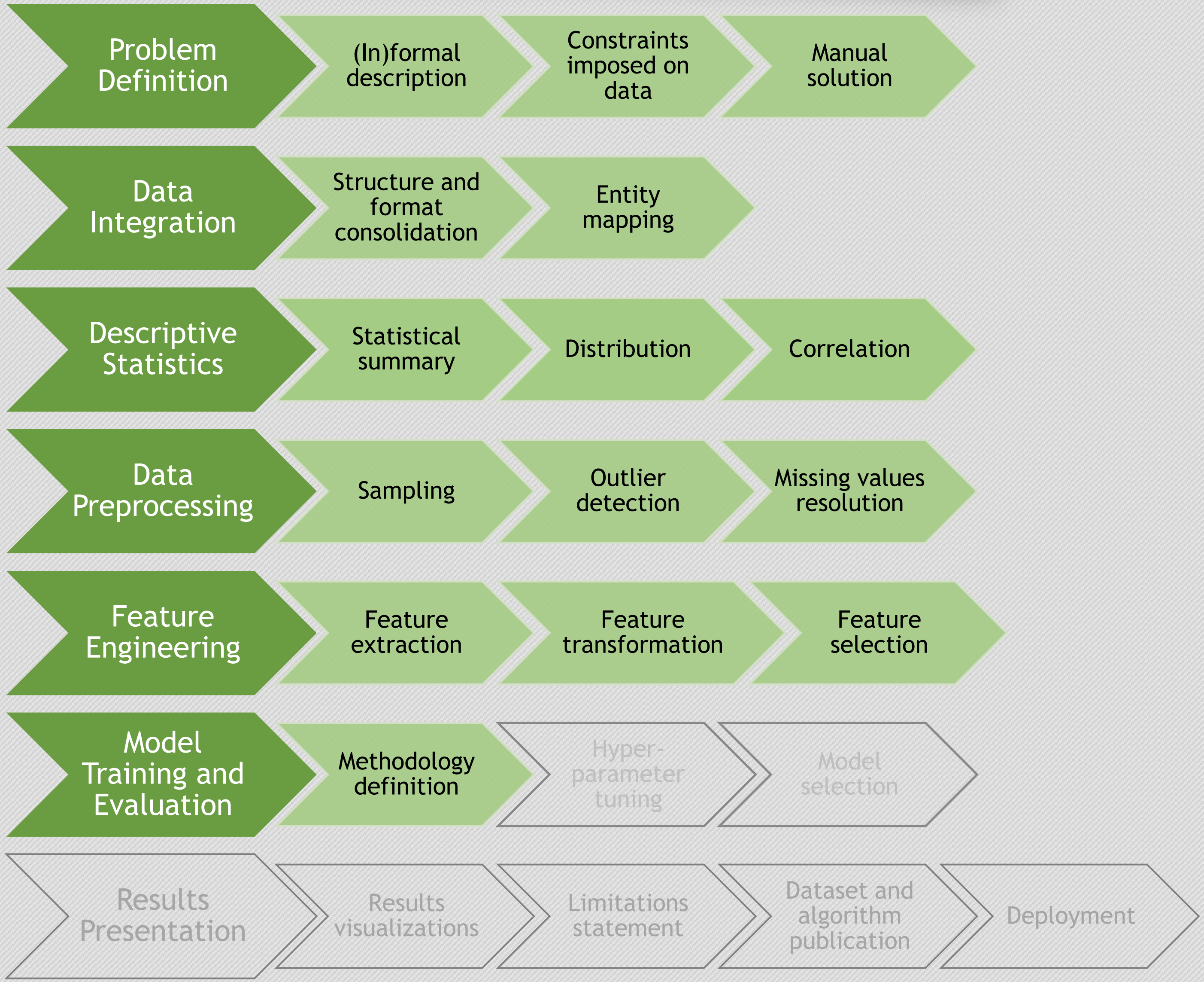
- Feature construction can lead to huge number of features
- Techniques
  - Filter methods
  - Wrapper methods
  - Embedded methods
- Hints
  - In the first iterations, use ML algorithms which have feature selection build-in (e.g. decision trees, random forest)



Methodology definition

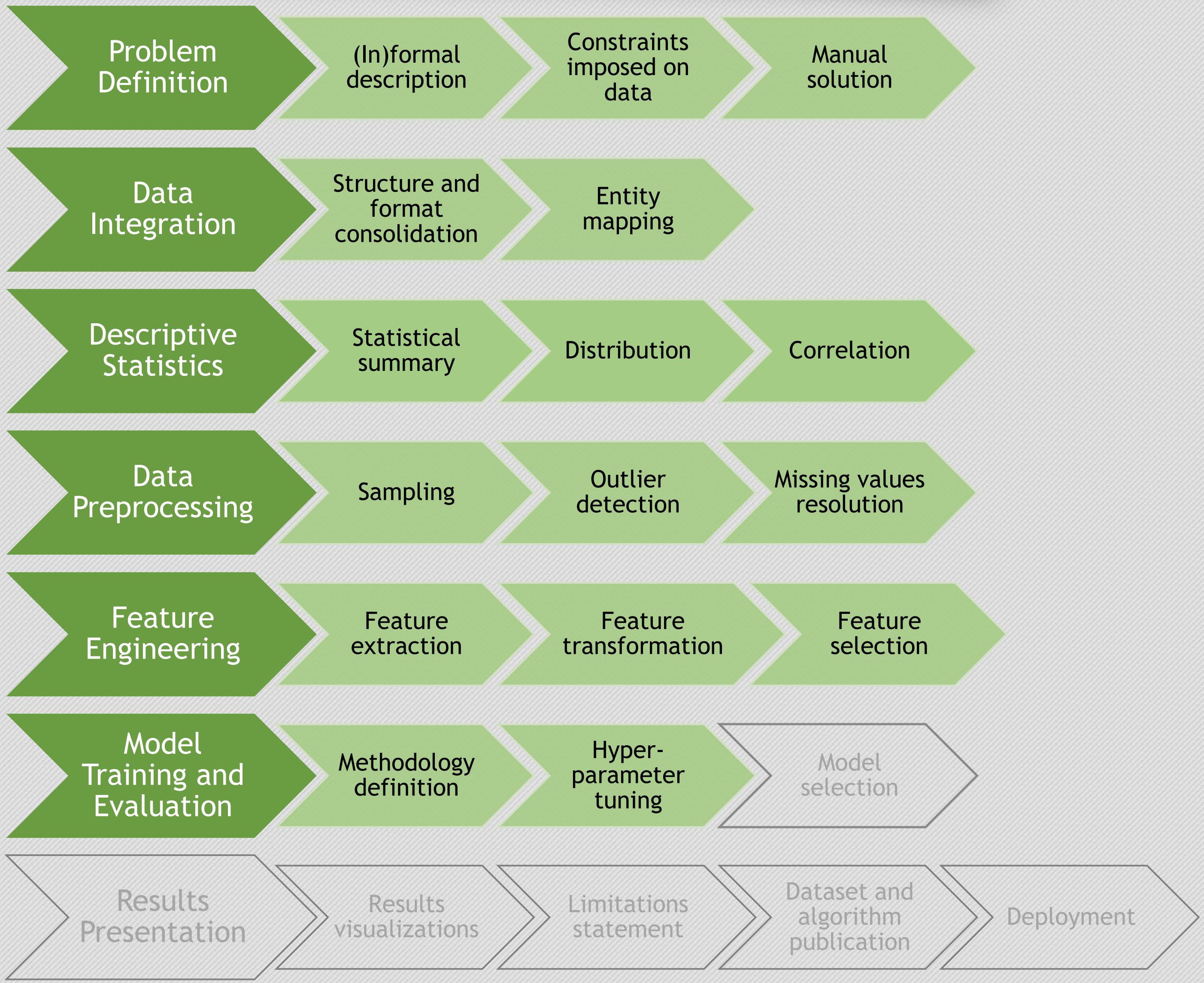
- Methodology definition contains
  - Evaluation steps
  - Metrics
  - Baseline

- Hints
  - Explicitly state your methodology
  - Select and define metrics suitable for your ML task
  - Distinguish training, testing and validating sets
  - Use cross-validation if necessary
  - Select appropriate baseline



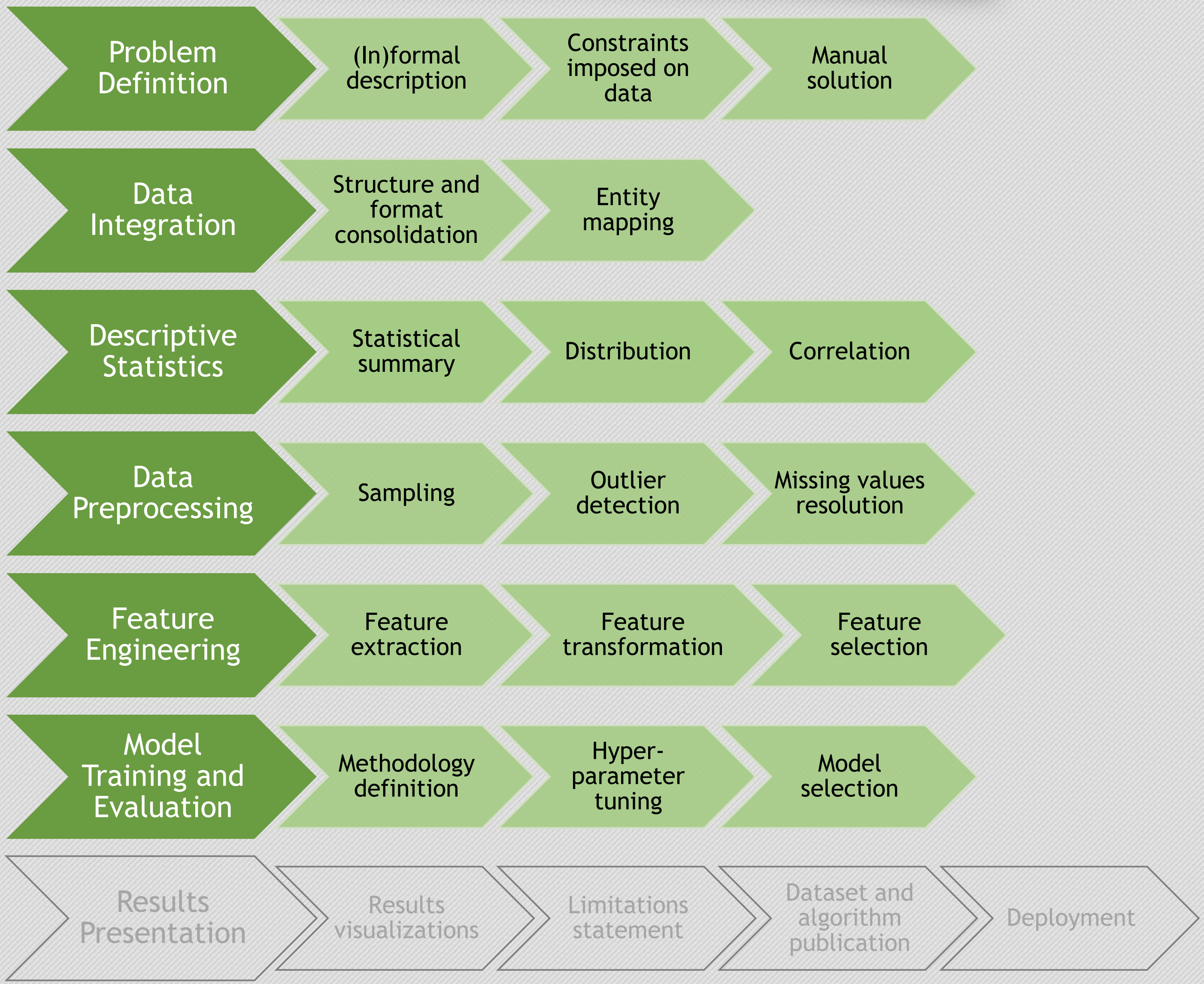
Hyperparameter tuning

- All ML algorithms required to adjust a set of hyperparameters
- Techniques
  - Grid-search, random-search, …
- Hints
  - Only in the very first iteration, you can rely on default algorithm parameters (but you need to know them)
  - In the next iterations, always do hyperparameter tuning



Model selection

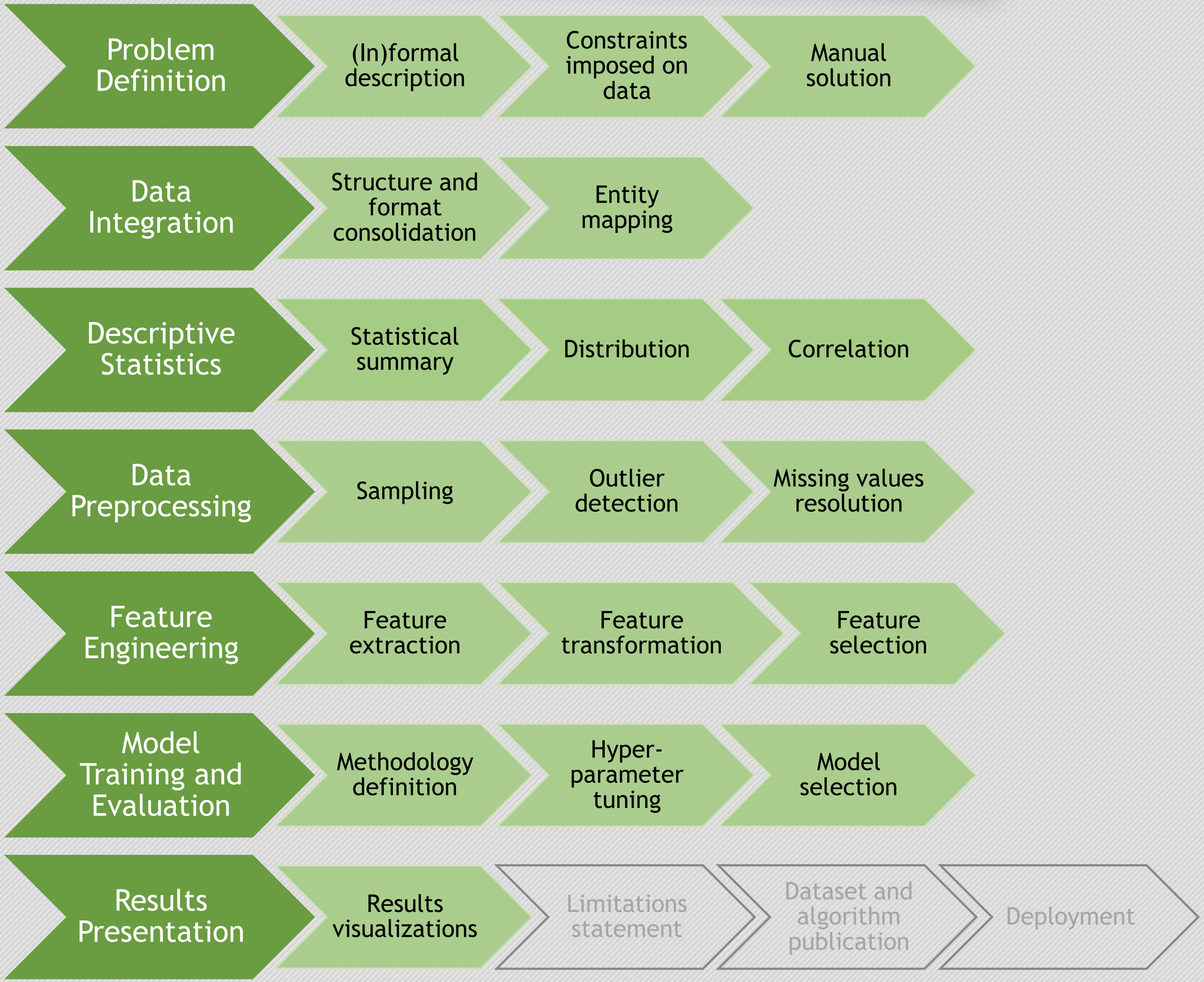
- The best model according to your stated problem/goal need to be selected
- Hints
  - Be aware that different models can perform better in different use cases



Results visualizations

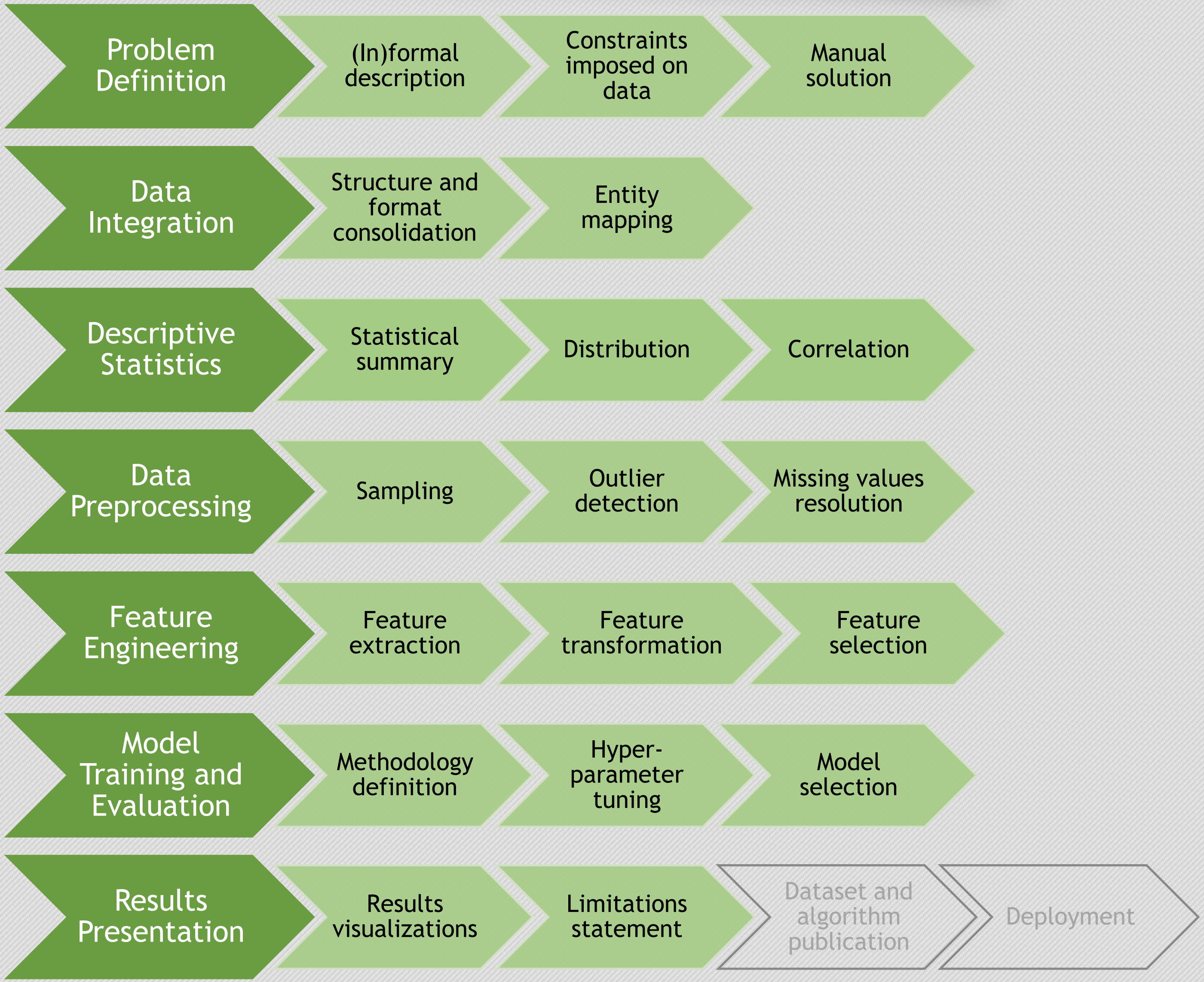
- Techniques
  - Tables
  - Charts
  - Jupiter Notebooks (Python, R, …)

- Hints
  - Primarily compare results answering your stated hypothesis
  - Secondary compare results for various ML algorithms, feature sets, …
  - Always provide a discussion on results



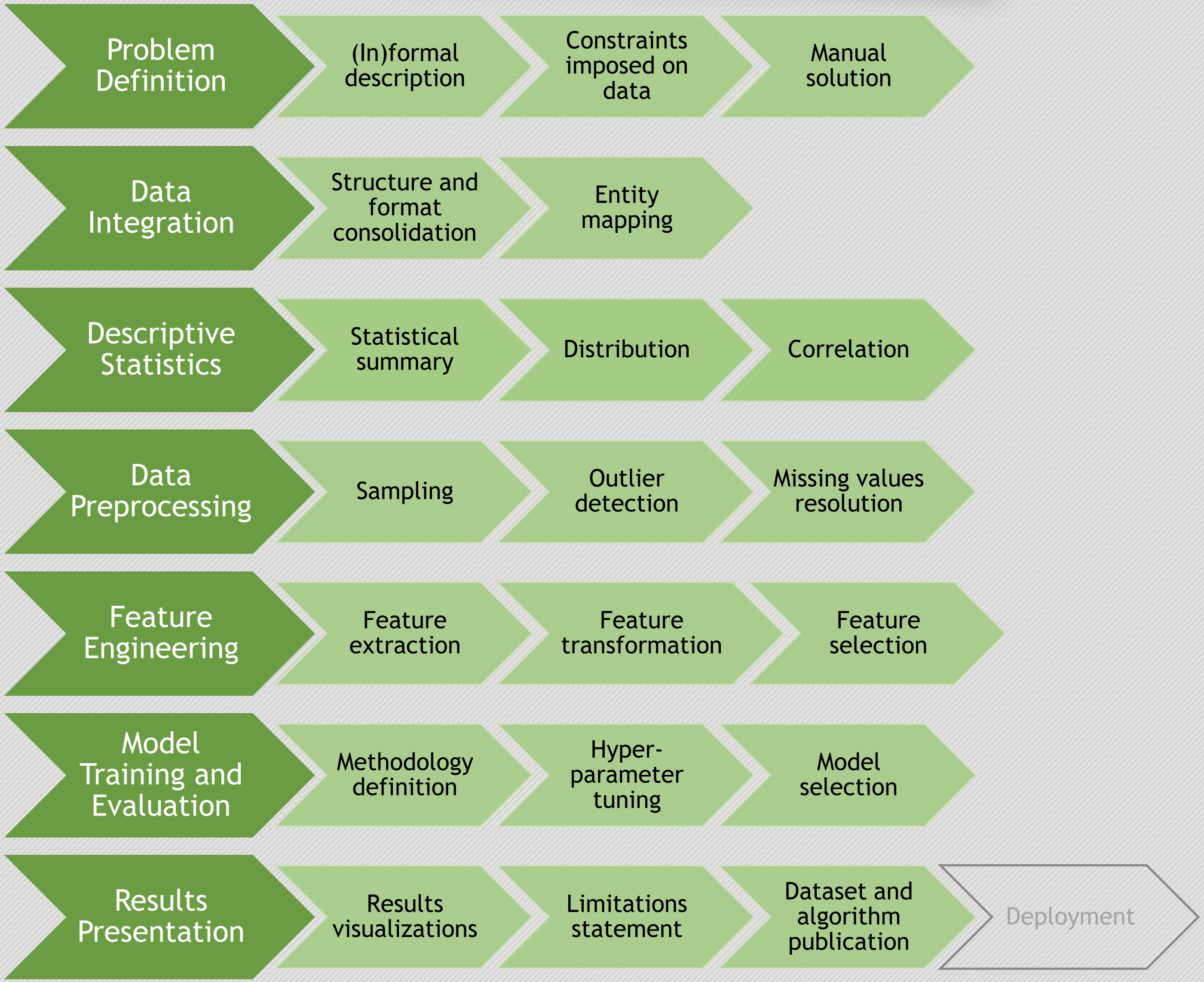
Limitation statement

- Hints
  - Always admit limitations of your methods
  - Discuss model transferability



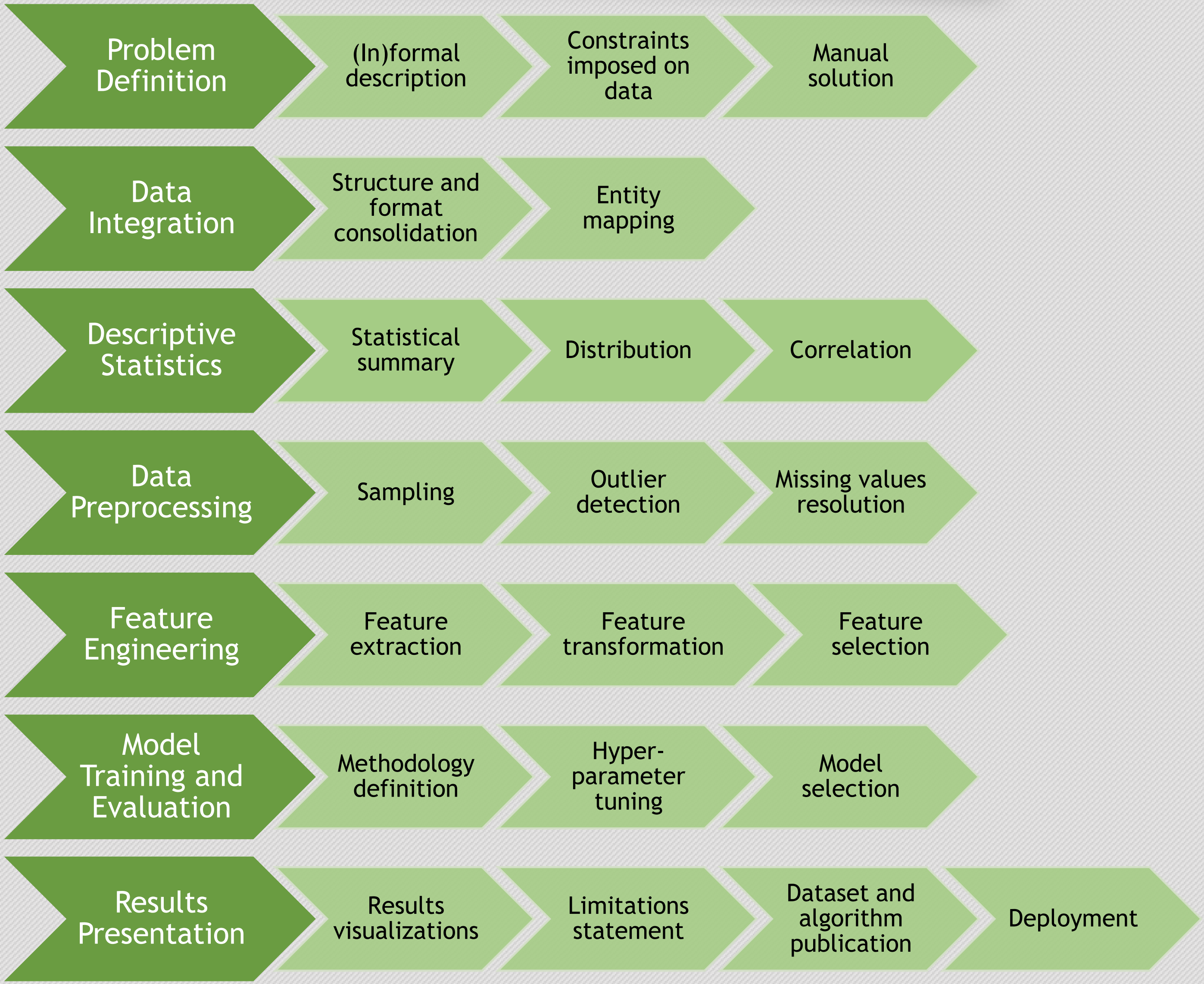
Dataset and algorithm publication

- Best results are 100% reproducible
- Hints
  - Make algorithm easily runnable
  - Describe steps how to rerun the evaluation
  - If possible, publish your dataset and algorithm (e.g. Github + Jupiter Notebook)



Deployment

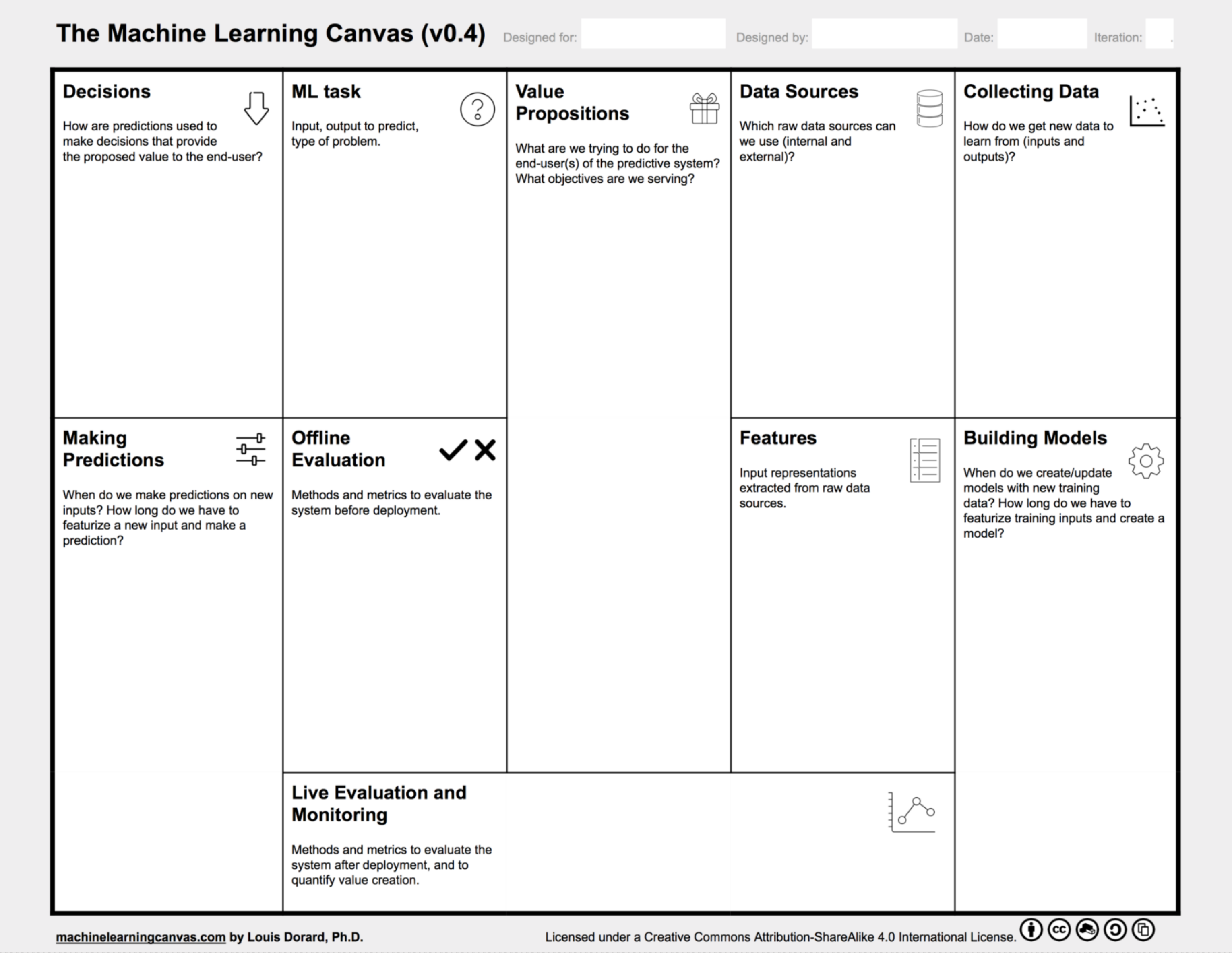
- Hints
  - If applicable, deploy the algorithm online, measure its performance and continue improving
  - If not, describe typical use cases



ML Workflow: Thesis sections’ mapping

Proposal	Problem Definition	(In)formal description	Constraints imposed on data	Manual solution	
Proposal + Implementation	Data Integration	Structure and format consolidation	Entity mapping		
Evaluation / Proposal	Descriptive Statistics	Statistical summary	Distribution	Correlation	
Evaluation / Proposal	Data Preprocessing	Sampling	Outlier detection	Missing values resolution	
Proposal + Implementation	Feature Engineering	Feature extraction	Feature transformation	Feature selection	
Evaluation	Model Training and Evaluation	Methodology definition	Hyper-parameter tuning	Model selection	
Evaluation	Results Presentation	Results visualizations	Limitations statement	Dataset and algorithm publication	Deployment

Machine Learning Canvas



The Machine Learning Canvas (v0.4)

Designed for: Designed by: Date: Iteration:

Decisions
How are predictions used to make decisions that provide the proposed value to the end-user?

ML task
Input, output to predict, type of problem.

Value Propositions
What are we trying to do for the end-user(s) of the predictive system? What objectives are we serving?

Data Sources
Which raw data sources can we use (internal and external)?

Collecting Data
How do we get new data to learn from (inputs and outputs)?

Making Predictions
When do we make predictions on new inputs? How long do we have to featurize a new input and make a prediction?

Offline Evaluation
Methods and metrics to evaluate the system before deployment.

Features
Input representations extracted from raw data sources.

Building Models
When do we create/update models with new training data? How long do we have to featurize training inputs and create a model?

Live Evaluation and Monitoring
Methods and metrics to evaluate the system after deployment, and to quantify value creation.

machinelearningcanvas.com by Louis Dorard, Ph.D.

http://machinelearningcanvas.com

ML Workflow: Typical Problems

- Curse of dimensionality
  - Dimensionality reduction, …
- Feature explosion
  - Feature selection, …
- Overfitting
  - Regularization parameters, pruning decision trees, …

ML Workflow: Typical Problems

- Imbalanced datasets
  - Under-sampling, over-sampling, different weights
- Small datasets
  - Cross validation, …
- Concept drift
  - Retrain model regularly, decaying factors, …

Conclusion

- Feature engineering is crucial
- Pay attention to describe and discuss results
- Be aware of the typical problems

- Take advantage of an opportunity to present your results, problems at Datalys
  - Reserve your slot in the Google doc