MODEL DEVELOPMENT PROCESS

A PREPRINT

Przemyslaw Biecek
Warsaw University of Technology & University of Warsaw
przemyslaw.biecek@gmail.com

July 10, 2019

ABSTRACT

Predictive modeling has an increasing number of applications in various fields. High demand for predictive models drives creation of tools that automate and support work of data scientist on the model development. To better understand what can be automated we need first a description of the model life-cycle.

In this paper we propose a generic Model Development Process (MDP). This process is inspired by Rational Unified Process (RUP) which was designed for software development. There are other approached to process description, like CRISP DM or ASUM DM, in this paper we discuss similarities and differences between these methodologies.

We believe that the proposed open standard for model development will facilitate creation of tools for automation of model training, testing and maintaining.

Keywords Automation · Process · Machine Learning · Predictive modeling · Data science

1 Introduction

High demand for predictive models drives creation of tools that automate and support model development, like H2O [H2O, 2019], DataRobot [DataRobot, 2019], mljar [Ploski, 2019], mlr [Bischl et al., 2016], tpot [Olson et al., 2016], tidymodels [Kuhn and Hadley, 2018], scikitlearn [Pedregosa et al., 2011] or others. To better understand the offering of such tools we need better description of the process of model development itself.

One of the most known approach to standardization of data mining projects is the CRISP DM methodology (Cross-industry standard process for data mining) [Chapman et al., 1999] presented in Figure 1. CRISP DM was conceived by five companies: Integral Solutions Ltd (ISL), Teradata, Daimler AG, NCR Corporation and OHRA, an insurance company [Wikipedia, 2019].

The key component of this approach is the break down of the whole process into six phases: business understanding, data understanding, data preparation, modeling, evaluation and deployment. In general these phases are performed in a consecutive manner, but is it allowed to return to previous phases. The process is iterative as the experience from previous phases helps in consecutive iterations.

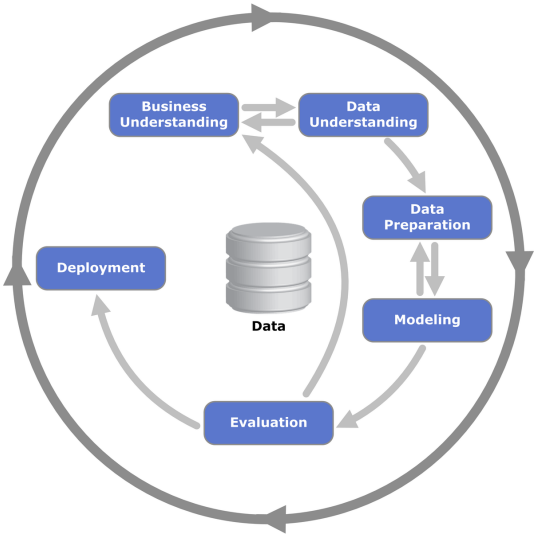


Figure 1: Diagram for *CRISP DM: Cross-industry standard process for data mining* https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining.

CRISP DM is a tool agnostic procedure. One of its founders, The ISL company, was later acquired by SPSS which was later acquired by IBM. IBM created a new methodology based on CRISP DM called Analytics Solutions Unified Method for Data Mining/Predictive Analytics (ASUM-DM see Figure 2). Other companies have their own methodologies, for example SAS has SEMMA.

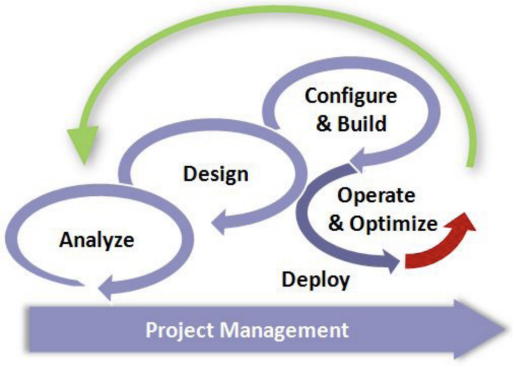
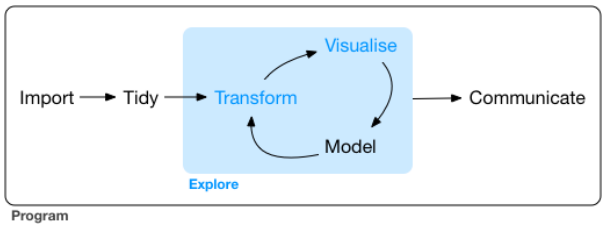


Figure 2: Diagram for *ASUM-DM: Analytics Solutions Unified Method* `ftp://ftp.software.ibm.com/software/data/sw-library/services/ASUM.pdf`.

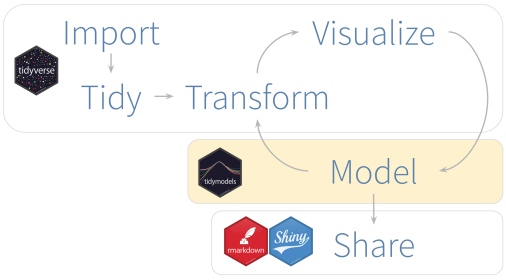
It should be noted that these methods have very wide range of applications, which covers almost any data driven problem, like segmentation or rule based decision systems. In next sections we will focus more on methodologies oriented on development of predictive models.

CRISP and ASUM are process oriented methodologies. Other approaches cover also methodologies that are more programming oriented. Probably one of the most popular is the approach proposed by Garrett Grolemund and Hadley Wickham in the *R for Data Science*, summarized in Figures 3a and 3b. This proposition is closely linked with the set of tools developed by RStudio in the tidyverse.

This approach highlights the iterative nature of data exploration with repetitive steps: Visualise, Model and Transform. The simplicity of this approach is tempting, yet one needs to remember that in parallel to programming effort we need to think also about other factors like documentation, validation, diagnostics and others.



(a) Diagram from *R for Data Science*, Garrett Grolemund, Hadley Wickham, `https://r4ds.had.co.nz/`



(b) Diagram from *A gentle introduction to tidymodels* `https://rviews.rstudio.com/2019/06/19/a-gentle-intro-to-tidymodels/`.

Figure 3: Diagrams for model development from Tidyverse.

Figure 4 comes from the book *Feature Engineering and Selection* by Max Kuhn and Kjell Johnson. It highlights the iterative feedback loop for model fitting and testing.

In all four presented approaches the iterative nature of the process is highlighted. To some degree this is fully justified as the work with data is almost always iterative as we know more and more and we can use the new knowledge in consecutive iterations. What may be confusing if the composition of consecutive iterations, as it's not the *iterate until convergence* procedure with similar steps repeated before some convergence criteria are met. In fact in data driven tasks consecutive steps have different compositions as some decision is being held and some forking paths are being closed.

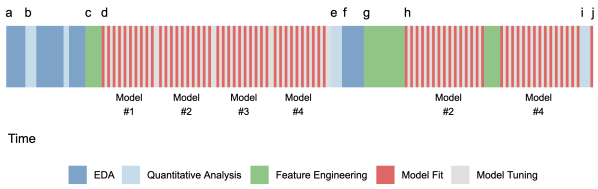


Figure 4: Diagram from *Modeling process according to FES* `https://bookdown.org/max/FES/`.

A different approach, more detailed and human centered is presented in the XAI misconceptions paper (see Figure 5). There are at least three interesting elements in this diagram. The process is designed explicitly for development of predictive models. Points that require human supervision are directly highlighted (like human review, debugging, explanations). The process is iterative but it is explicit what drives next iterations, e.g. improvements in accuracy, fairness or transparency.

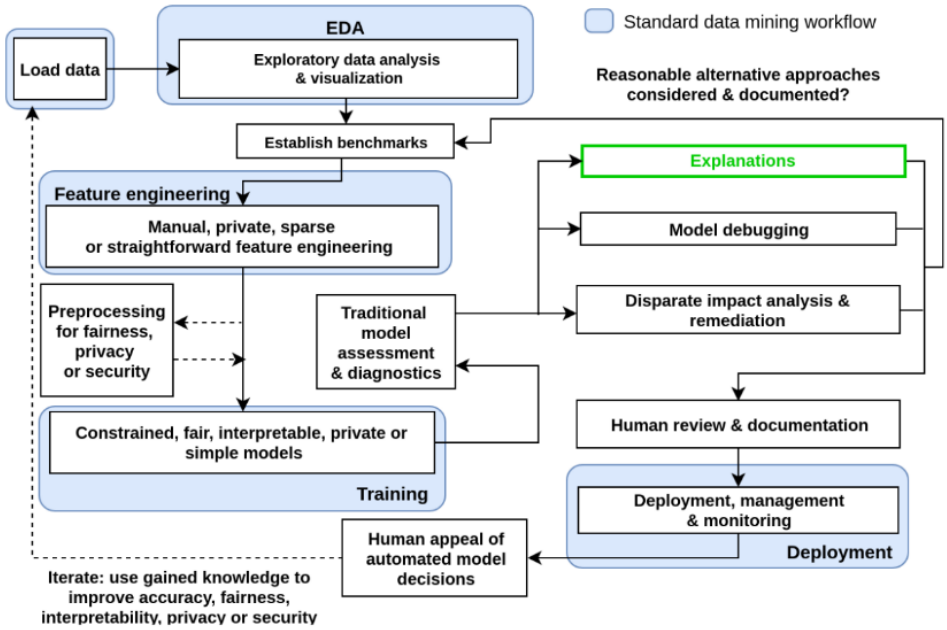
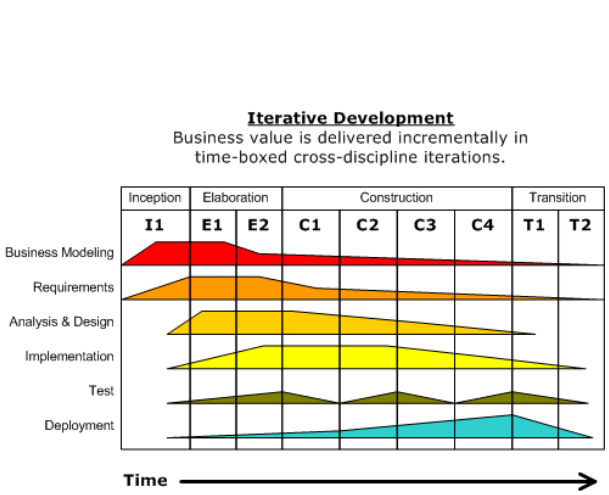
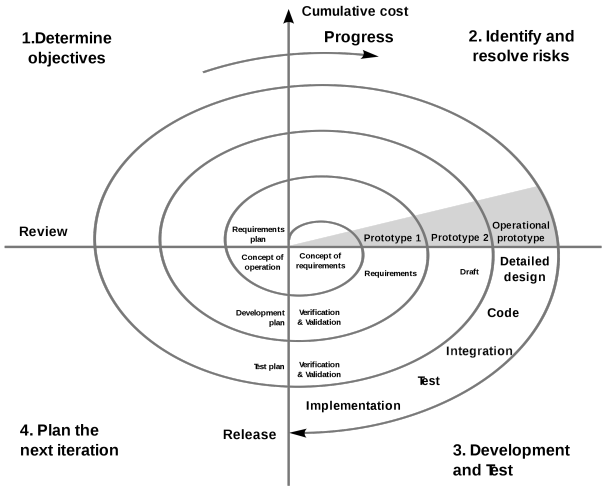


Figure 5: Diagram from *On Explainable Machine Learning Misconceptions and A More Human-Centered Machine Learning*, Patrick Hall, `https://github.com/jphall663/xai_misconceptions/blob/master/xai_misconceptions.pdf`.



(a) Diagram for *Rational Unified Process* `https://en.wikipedia.org/wiki/RUP_hump`



(b) Diagram for *Spiral model (Boehm, 1988)* `https://en.wikipedia.org/wiki/Spiral_model`

Figure 6: Methodologies for software development .

It is interesting to compare these approaches with similar yet more mature field of software development. There are many methodologies that are preferred by different companies depending on scale and nature of the process. Two

interesting solutions are the Rational Unified Process [Kruchten, 1998, Jacobson et al., 1999] (see Figure 6a) developed by the Rational company and the Spiral model [Boehm, 1988] (see Figure 6b). They replaced the Waterfall model as the iterative nature with quick updated and early checks help to limit number of problems with changing requirements or changes in architecture.

What is interesting in both these proposition is that the process is iterative, but clearly consecutive iterations build on top of previous one. To make it possible, each iteration must be supplemented with some documentation that track the progress of the whole process.

2 Model Development Process (MDP) in details

MDP process serves as a skeleton for the `http://DrWhy.AI` [Biecek, 2018] set of tools. This proposition is based on Rational Unified Process and is adapted for development of predictive models.

The described structure is generic. Specific projects may require some changes and adaptations.

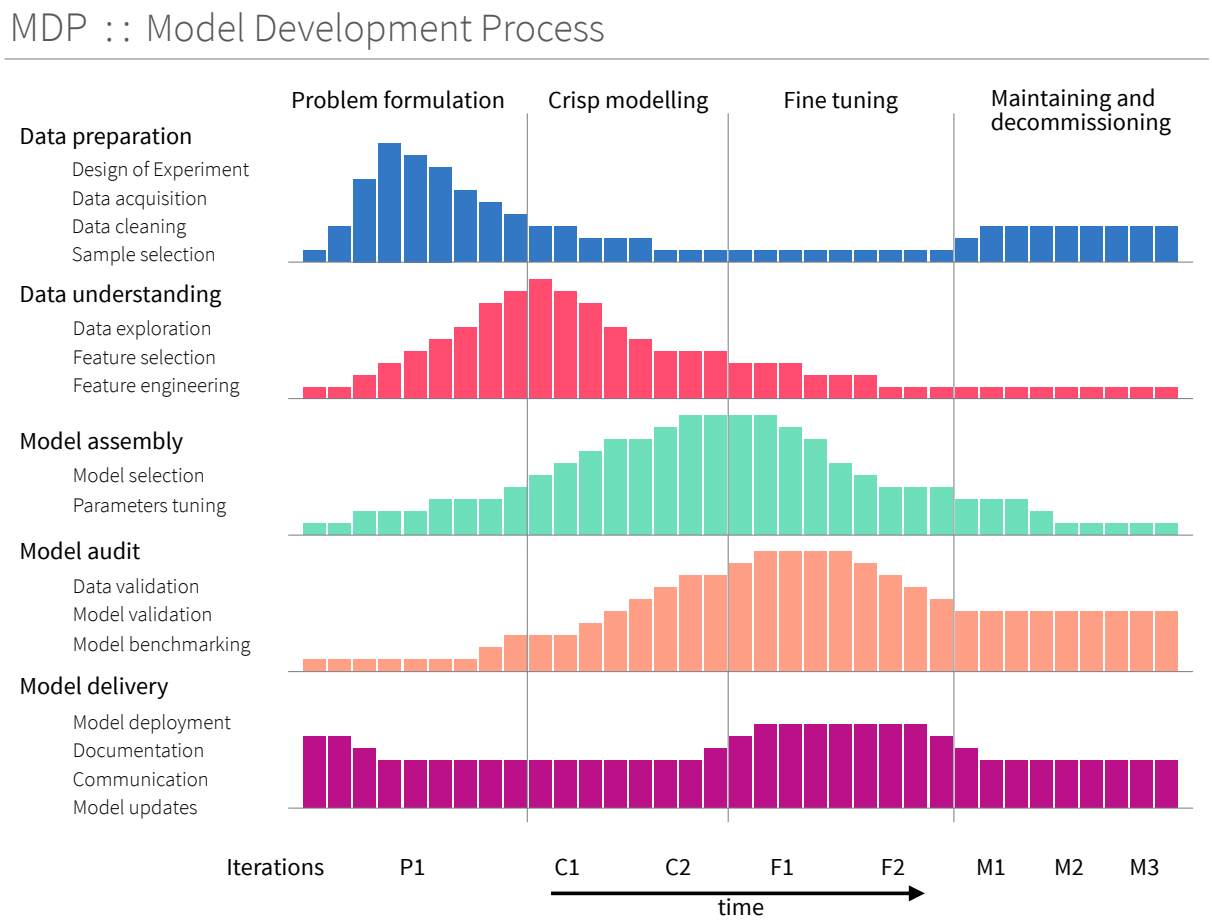


Figure 7: General structure of the *Model Development Process*. Columns correspond to the consecutive phases of model construction. Each phase builds on the knowledge gathered from previous phases. Rows correspond to specific activities that are to be performed in each phase. Heights of bars correspond to the importance of specific activity in each phase.

The process is divided into four phases of model life cycle. These phases are used to trace the progress in the model development. As in other methodologies MDP is based on series of iterations. Each phase may be composed out of one or more iterations.

Tasks that are to be performed in each iteration are listed in rows. As the knowledge about the problem increases every iteration, different tasks require more attention.

2.1 Phases and iterations

Process of model development is divided into four phases of model life cycle from the conception, assembly, tuning till the production.

- Problem formulation — What we want to improve? How we measure the improvement?. The goal of this phase is to precisely define needs for the predictive models, write down *definition of done* and define data set that will be used for training and validation. After this phase we know which performance measures will be used for the assessment of the final model.
- Crisp modeling — Generate a prototype quickly. Make sure that this is what client really wants.. The goal of this phase is to validate the *definition of done*. Here we create first versions of models in order to better understand how close we are to the desired solution.
- Fine tuning — Improve the prototype to maximize the desired target measure/minimize loss function.. The goal of this phase is to tune models identified in the previous phase. Usually in this phase we create large collection of models in order to select the best one (according to some metrics).
- Maintaining and decommissioning — How model works in the production environment? How it will be updated and how to monitor it's accuracy and fairness. Prepare for model decommissioning one the performance drops too much.. Developed model go for production. The goal of this phase is to monitor the model and make sure that model performance have not degenerated. Every model will be outdated some day, prepare for the end of model life cycle.

In more complex projects one phase may be divided into set of iterations. The maintaining phase usually is composed out of series of periodic health-checks.

2.2 Tasks

Phases corresponds to the general progress in model development, while tasks corresponds to programming and analytic activities that needs to be done in each iteration. Importance of particular tasks is changing along model life-cycle. In the diagram we showed some general patterns, but specific problems may require more effort in some of these phases.

- Data preparation — Activities needed for selection of the training, test and validation data.
  - Data acquisition. Sometimes data needs to be read from file, from database or from some stream. Sometimes we need to scrap data from website. Sometimes one datset is not enough and we need to acquire more (maybe paid) datasets that will be combined.
  - Data cleaning. Different data sources have different quality. Sometimes some values needs recoding, errors in the data needs to be spotted.
  - Sample selection. Good model requires good and carefully selected dataset. Outliers needs to be handled. If data is not balanced or is heterogeneous this needs to be handled, typically through oversampling, undersampling or segmentation.
- Data understanding — Activities needed for getting some lever of familiarity with the data, needed for further modeling.
  - Data exploration. What are uni- and multi- variate distributions. What are relation between dependent variable and explanatory variables. Do we have missing values. How strong is the correlation between different features.
  - Feature selection. Which variables shall be included in the model. Assessment of their predictive power independently and in groups of other variables.
  - Feature engineering. How variables should be encoded. Factors may need some recoding, continuous variables may need some transformations of discretisation. Groups of variables may need blending.
- Model assembly — Activities needed for the construction of the model.
  - Model selection. There is an increasing number of different procedures for model construction. Further, new models can be created as a combination of other models.
  - Parameters tuning. Most procedures for model constructions are parametrized. Different strategies may be employed to identify best set of parameters.
- Model audit — Activities needed for monitoring model performance, fairness and stability.
  - Data validation. Is there a change in the structure of the data, distributions of variables or relation structure?

  - Model validation. Is there a change in model performance between training, test and validation data. Is there a change in performance in the new batch of validation data? Is there any issue in model fairness?
  - Model benchmarking. How good is a given model in comparison to other models?
- Model delivery — Activities needed for model release.
  - Model deployment. Model needs to be put in the production environment keeping same version of dependent libraries.
  - Documentation. Decisions that lead to the final model needs to be saved. Model and data used for training should be clearly defined. Documentation shall be gathered and expanded through the full model lifetime.
  - Communication. Reports, charts, tables, all artifact that are used to consult the model with the client in a easy to understand way.
  - Model updates. With new batches of data one may plan model retraining to adjust for recent data. This phase is more common for time-series models.

3 Acknowledgments

I would like to thank Patrick Hall, Max Kuhn, Wit Jakuczun and Alicja Gosiewska for valuable comments and discussions related to MDP.

References

[Biecek, 2018] Biecek, P. (2018). DALEX: Explainers for Complex Predictive Models in R. *Journal of Machine Learning Research*, 19(84):1–5.

[Bischl et al., 2016] Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., Casalicchio, G., and Jones, Z. M. (2016). mlr: Machine learning in r. *Journal of Machine Learning Research*, 17(170):1–5.

[Boehm, 1988] Boehm, B. (1988). *A Spiral Model of Software Development and Enhancement*.

[Chapman et al., 1999] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., and Wirth, R. (1999). *The CRISP-DM 1.0 Step-by-step data mining guide*.

[DataRobot, 2019] DataRobot (2019). *DataRobot: automated machine learning platform*.

[H2O, 2019] H2O (2019). *H2O: in-memory platform for distributed, scalable machine learning*.

[Jacobson et al., 1999] Jacobson, I., Booch, G., and Rumbaugh, J. (1999). *The Unified Software Development Process*.

[Kruchten, 1998] Kruchten, P. (1998). *The Rational Unified Process: An Introduction*.

[Kuhn and Hadley, 2018] Kuhn, M. and Hadley, W. (2018). *tidymodels: Easily Install and Load the 'Tidymodels' Packages*. R package version 0.0.2.

[Olson et al., 2016] Olson, R. S., Urbanowicz, R. J., Andrews, P. C., Lavender, N. A., Kidd, L. C., and Moore, J. H. (2016). *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings, Part I*, chapter Automating Biomedical Data Science Through Tree-Based Pipeline Optimization, pages 123–137. Springer International Publishing.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Ploski, 2019] Ploski, P. (2019). *mljar: automated machine learning platform*.

[Wikipedia, 2019] Wikipedia (2019). *CRISP DM: Cross-industry standard process for data mining*.