2487-S2 Machine Learning
(Data Science for Business 201)
Week 2: End-to-End Machine Learning Project

Qiwei Han, Ph.D.

Nova School of Business and Economics, Portugal

Masters Program in Economics, Finance and Management

February 8/9th, 2023

NOVA SCHOOL OF BUSINESS & ECONOMICS

Outline

- From business problem to machine learning problem: a recipe
- Standard data mining process

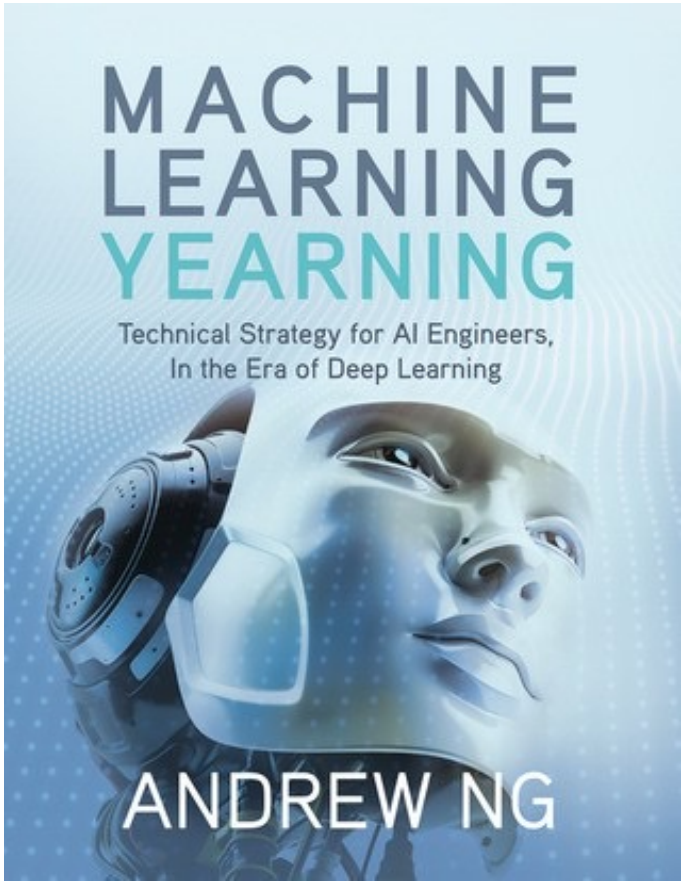
NOVA SCHOOL OF
BUSINESS & ECONOMICS

Rules of Machine Learning: Best Practices for ML Engineering

Martin Zinkevich

This document is intended to help those with a basic knowledge of machine learning get the benefit of best practices in machine learning from around Google. It presents a style for machine learning, similar to the Google C++ Style Guide and other popular guides to practical programming. If you have taken a class in machine learning, or built or worked on a machine-learned model, then you have the necessary background to read this document.

Rules of Machine Learning (RML)



Machine Learning Yearning (MLY)

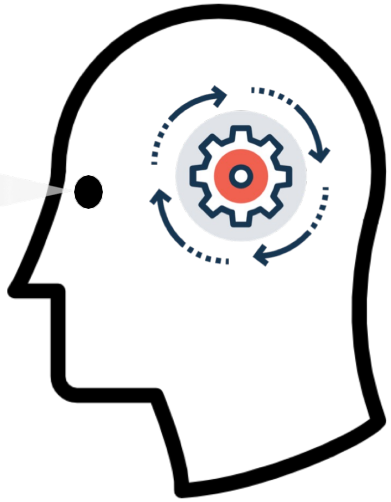
NOVA SCHOOL OF BUSINESS & ECONOMICS

When should you use Machine Learning?

From business problem to machine learning problem: a recipe

Step-by-step “recipe” for qualifying a business problem as a machine learning problem

1. Do you need machine learning?
2. Can you formulate your problem clearly?
3. Do you have sufficient data?
4. Does your problem have a regular pattern?
5. Can you find meaningful representations of your data?
6. How do you define success?



NOVA SCHOOL OF BUSINESS & ECONOMICS

When to use machine learning?

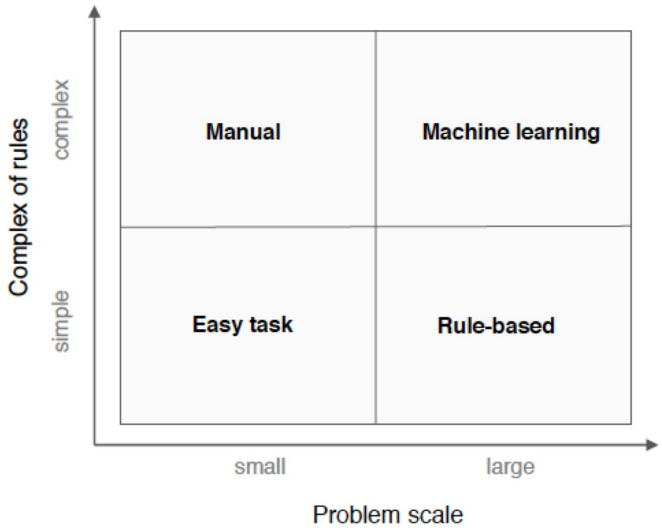
1

Do you need machine learning?

- Do you need to automate the task?
- High volume tasks with complex rules and unstructured data are good candidates

Example: sentiment analysis

- High volume of reviews on the Web
- Unstructured text
- Human language is complex and ambiguous



Problem formulation

2

Building a model is probably not the end goal

- How does company expect to use and benefit?

Can you formulate your problem clearly?

- What do you want to predict given which input?
- Pattern: ”given X, predict Y”
  - What is the input?
  - What is the output?

Example: sentiment analysis

- Given a customer review, predict its sentiment
  - Input: customer review text
  - Output: positive, negative, neutral

Collect Data

3

Do you have sufficient data?

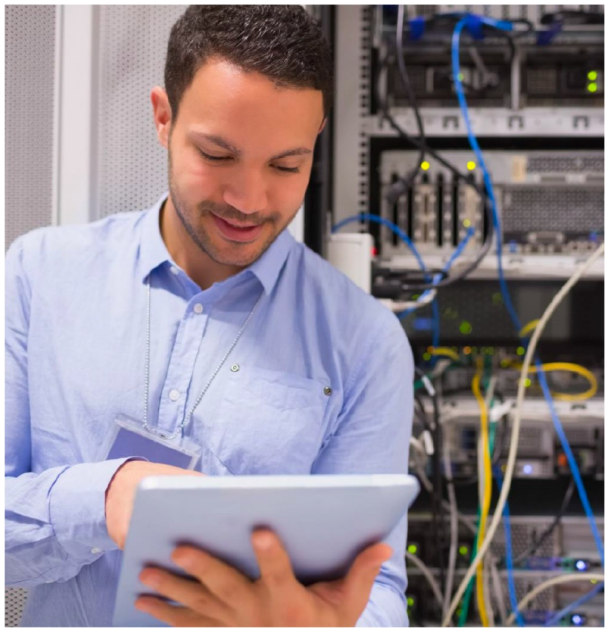
- Machine learning always requires data

Do you have the right data?

- Supervised learning task needs labeled data

Example: sentiment analysis

- Millions of customer reviews and ratings from the Web



Patterns in the data

4

Does your problem have a regular pattern?

- Machine learning learns patterns from the data
- Hard to learn patterns that are irregular or rare

Example: sentiment analysis

- Positive words like *good, awesome, or love* appear more often in highly-rated reviews
- Negative words like *bad, lousy, or disappointed* appear more often in poorly-rated reviews

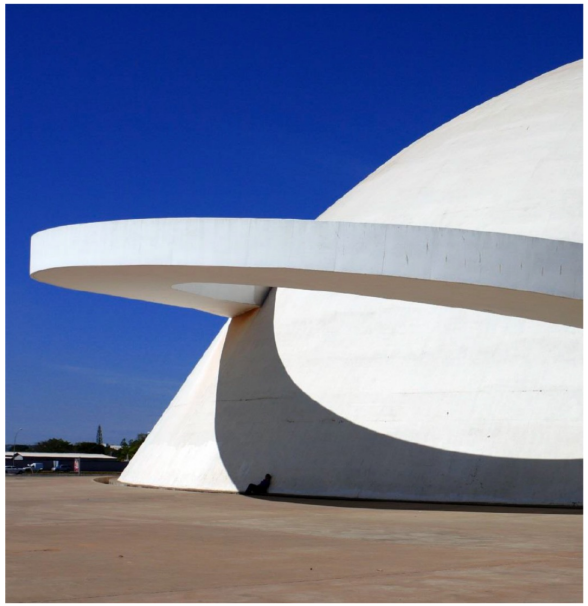
Representations and features

Can you find meaningful representations of the data?

- Machine learning models ultimately operates on numbers in feature vectors
- Engineering good features often determines the success of machine learning

Example: sentiment analysis

- Represent customer review as vector of word frequencies
- Label is positive (4-5 stars), negative (1-2 stars), neutral (3-stars)



Evaluate success

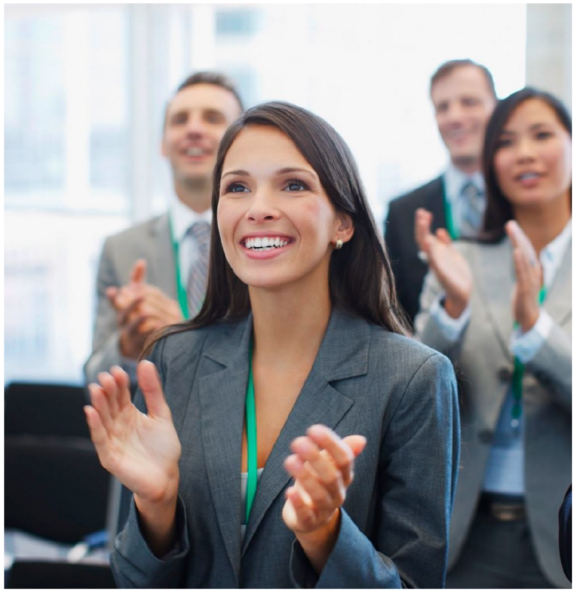
6

How do you define success?

- Machine learning optimize a training criteria (minimize errors)
- The evaluation function has to support the business goals

Example: sentiment analysis

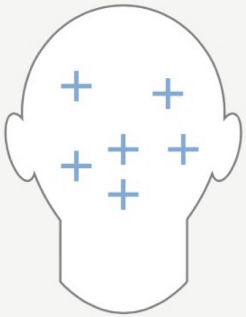
- Accuracy: percentage of correctly predicted labels



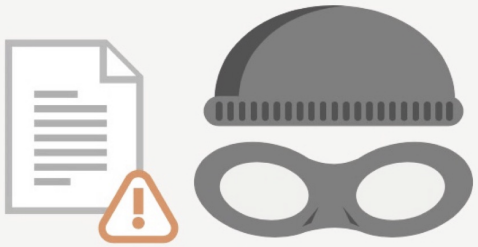
Wrap-up: When should you use machine learning

Consider using machine learning when you have a complex task or problem involving a large amount of data and lots of variables, but no existing formula or equation.
For example, machine learning is a good option if you need to handle situations like these:

Hand-written rules and equations are too complex—as in face recognition and speech recognition.



The rules of a task are constantly changing—as in fraud detection from transaction records.



The nature of the data keeps changing, and the program needs to adapt—as in automated trading, energy demand forecasting, and predicting shopping trends.

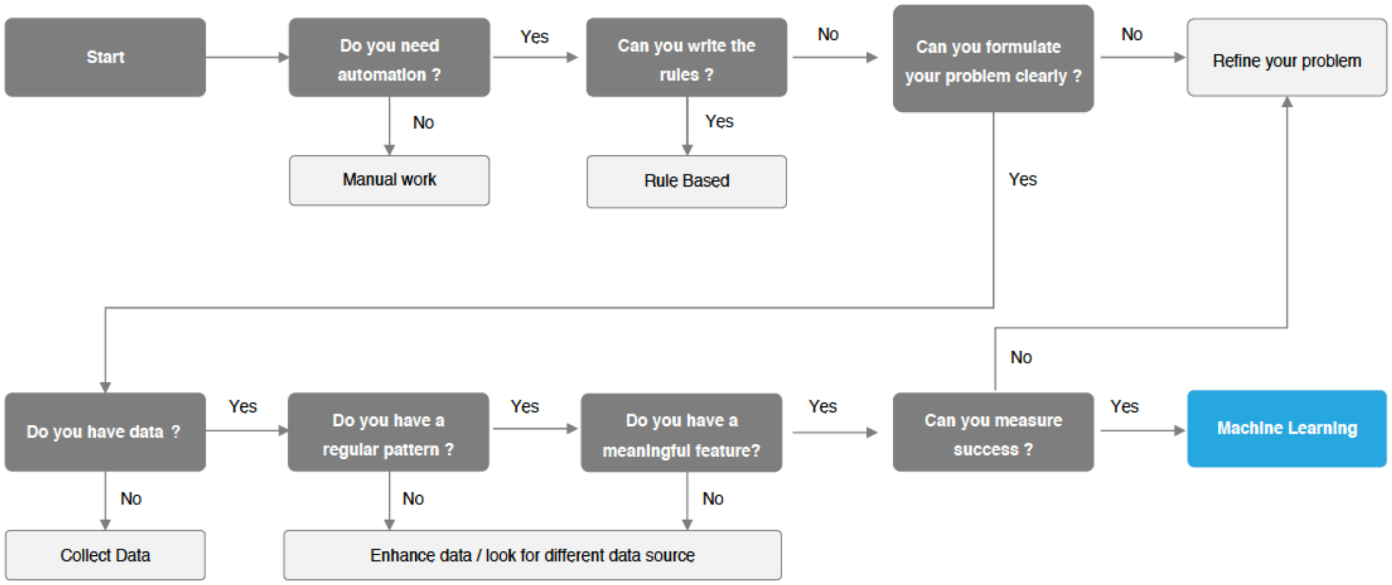
NOVA SCHOOL OF BUSINESS & ECONOMICS

Before Machine Learning

- **RML #1: Don't be afraid to launch a product without machine learning**
  - Manual job based on simple heuristic rules often work
  - Do not use machine learning until you have data
- **RML #2: First, design and implement metrics**
  - Check the historical data and design your task with metrics in mind
- **RML #3: Choose machine learning over a complex heuristic**
  - Once you have the data and a basic idea of your metric, move on to machine learning
  - Update and maintain machine learning model is much easier than complex heuristic

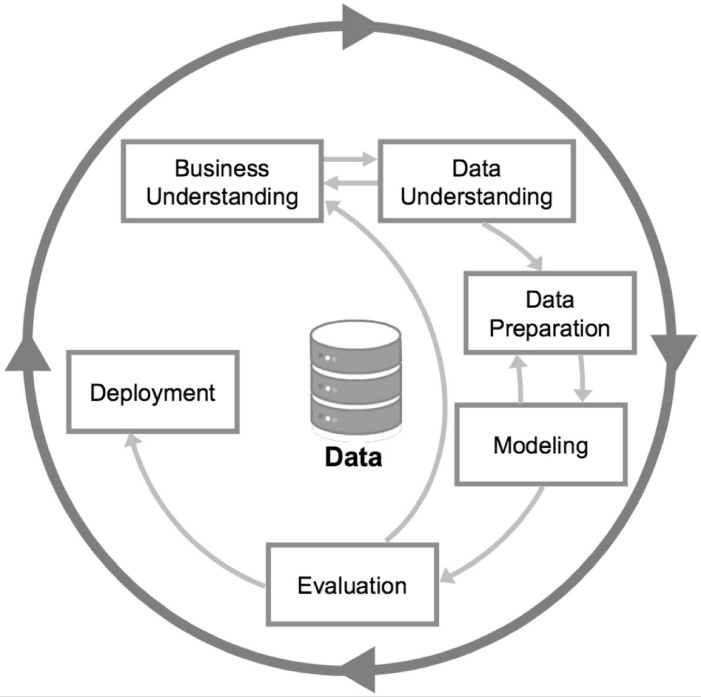
NOVA SCHOOL OF BUSINESS & ECONOMICS

From business problem to machine learning problem



NOVA SCHOOL OF BUSINESS & ECONOMICS

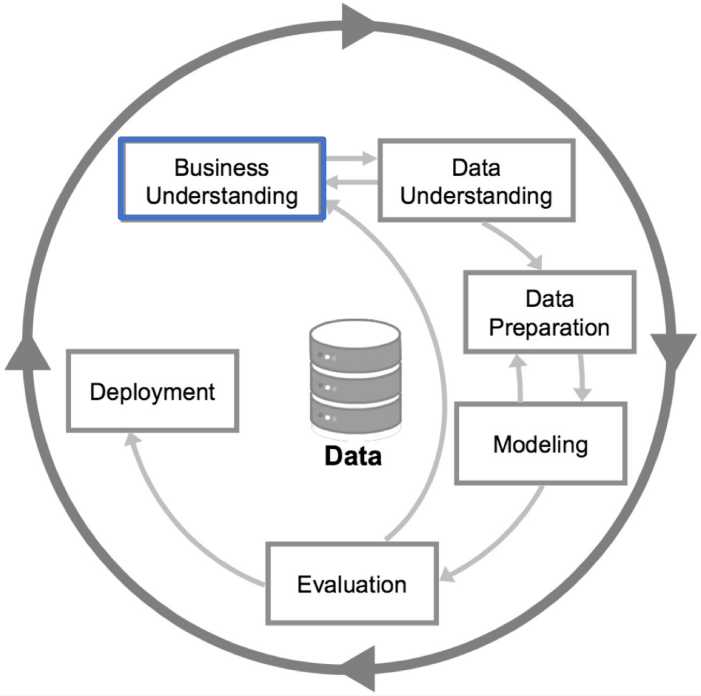
Standard Data Mining Process



Cross-industry standard process for data mining (CRISP-DM)

NOVA SCHOOL OF BUSINESS & ECONOMICS

Understand the business

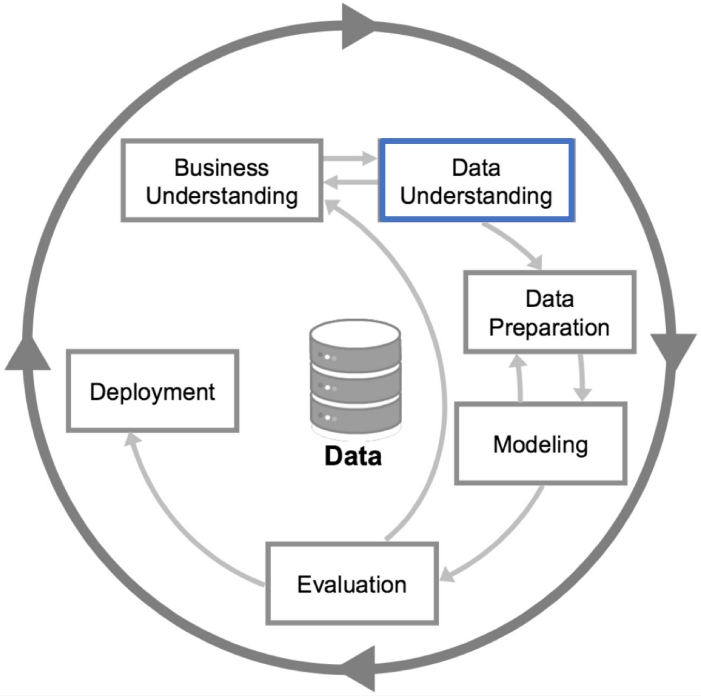


Get a clear understanding of the problem you're going to solve, how it impacts your organization, and your goals for addressing it. Tasks in this phase include:

- Identify your business goals
- Assessing your situations
- Defining your data science goals
- Producing your project plan

NOVA SCHOOL OF BUSINESS & ECONOMICS

Data Understanding



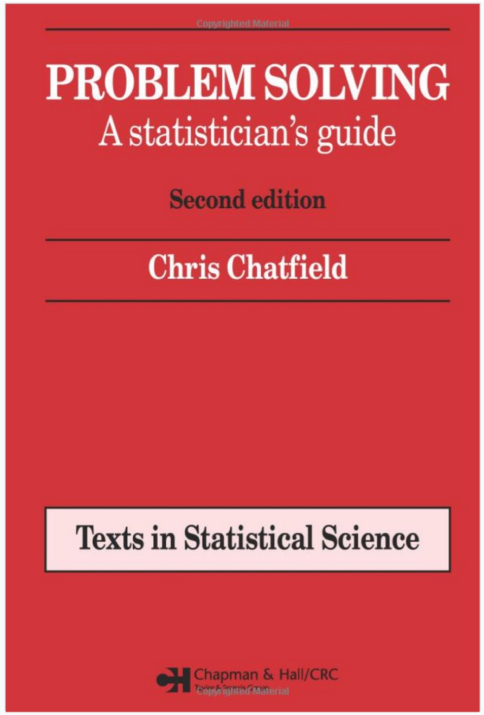
- Collect initial data
  - Access and load the data
  - Joining data from multiple sources into one dataset
- Describe data
  - Descriptive statistics
  - The structure of data
- Explore data
  - Data exploration offers an early view into the data
  - Visualizing the data to look for patterns in the data
  - A number of data issues can be uncovered during this step
  - Possibly formulate hypotheses that could lead to new data collection and experiments
- Verify data quality
  - Identify errors, outliers, and missing values

NOVA SCHOOL OF BUSINESS & ECONOMICS

Data Understanding

Explore Data – Initial Data Analysis (IDA)

- Initial data analysis (IDA) is an essential part of nearly every analysis. It includes analysis of:
  - The structure of data
  - The quality of data
  - Descriptive statistics
- The data are modified according to the IDA:
  - Adjust extreme observations
  - Estimate missing observations
  - Transform variables
  - Bin data
  - Create new variables



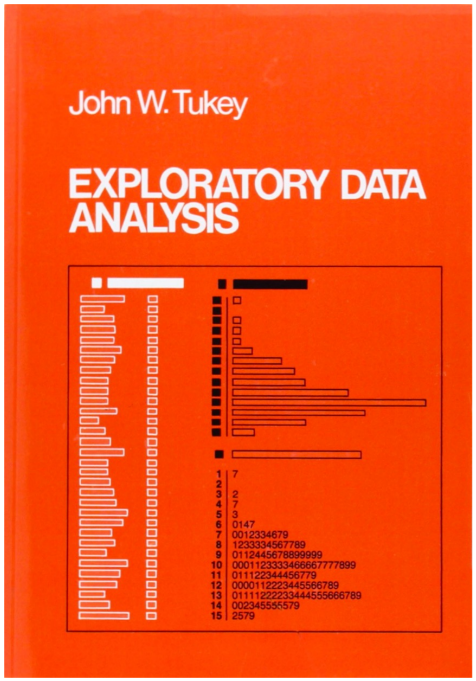
NOVA SCHOOL OF BUSINESS & ECONOMICS

Data Understanding

Explore Data – Explorative Data Analysis (EDA)

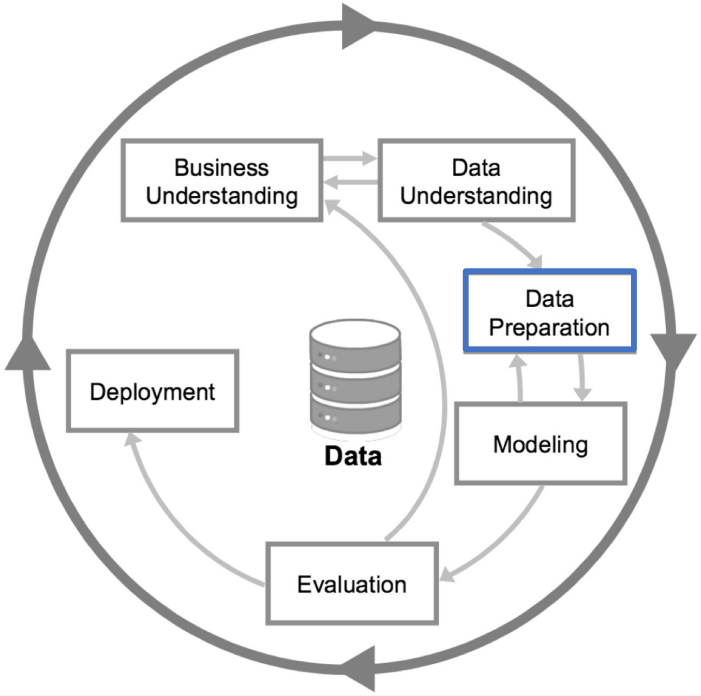
- EDA is an approach to analyzing data for the purpose of formulating hypotheses that are worth testing:
  - Data visualization is the common approach
  - EDA nowadays is mixed with IDA

- It is important to understand what you CAN DO before you learn to measure how WELL you seem to have it



NOVA SCHOOL OF BUSINESS & ECONOMICS

Data Preparation

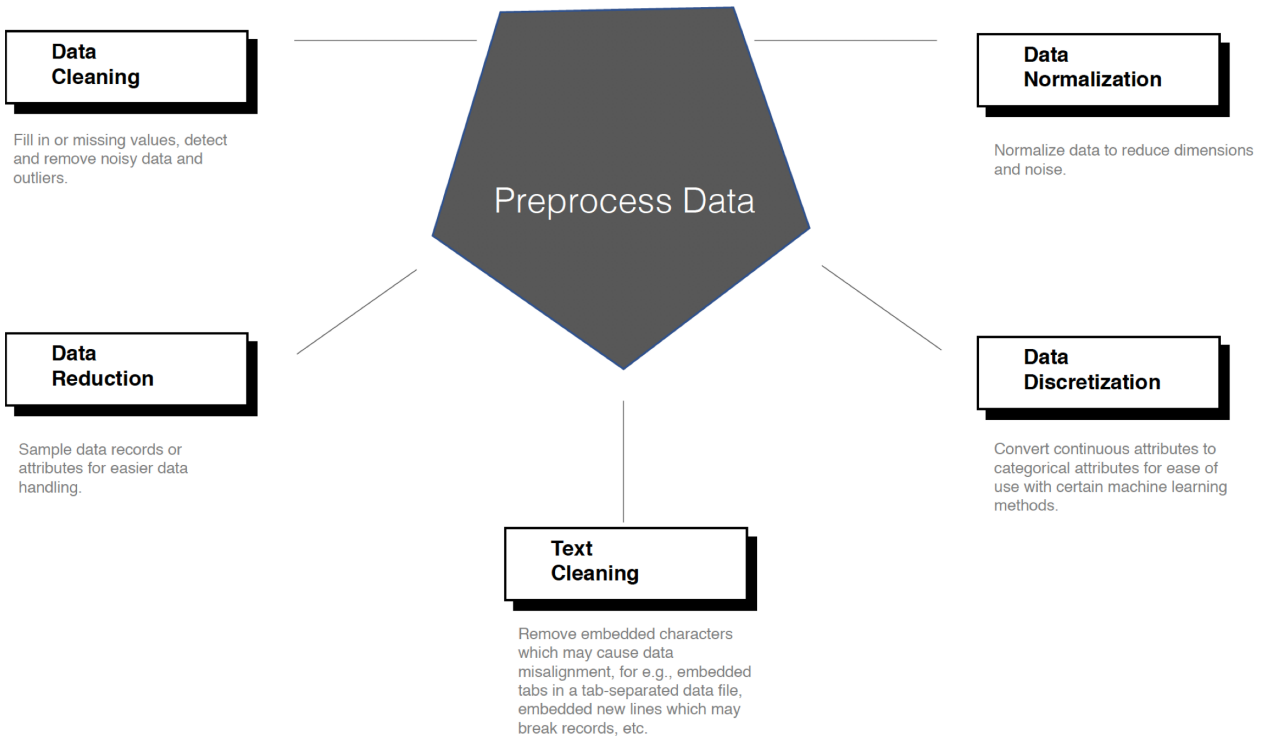


- Messy data is by far the most time-consuming aspect of the typical data scientist's work flow

- Data preparation/curation/preprocessing
  - Invalid values
  - Formats
  - Uniqueness
  - Missing values
  - Misspellings
  - Misfielded values

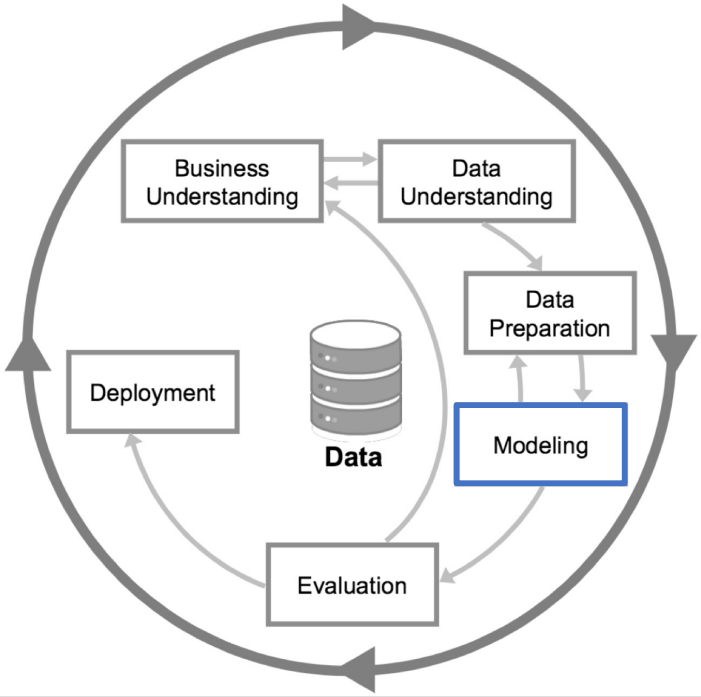
NOVA SCHOOL OF BUSINESS & ECONOMICS

Data curation



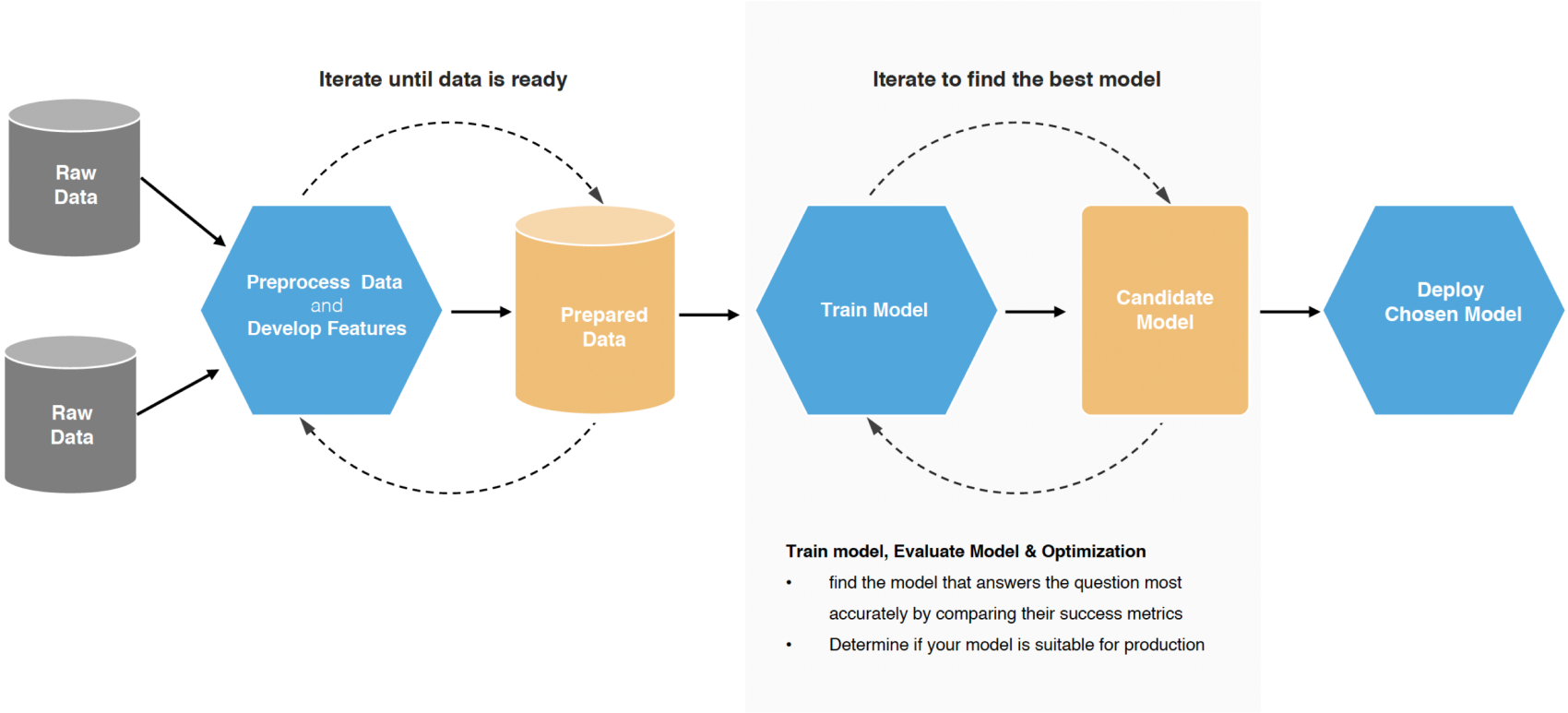
NOVA SCHOOL OF BUSINESS & ECONOMICS

Train the Model



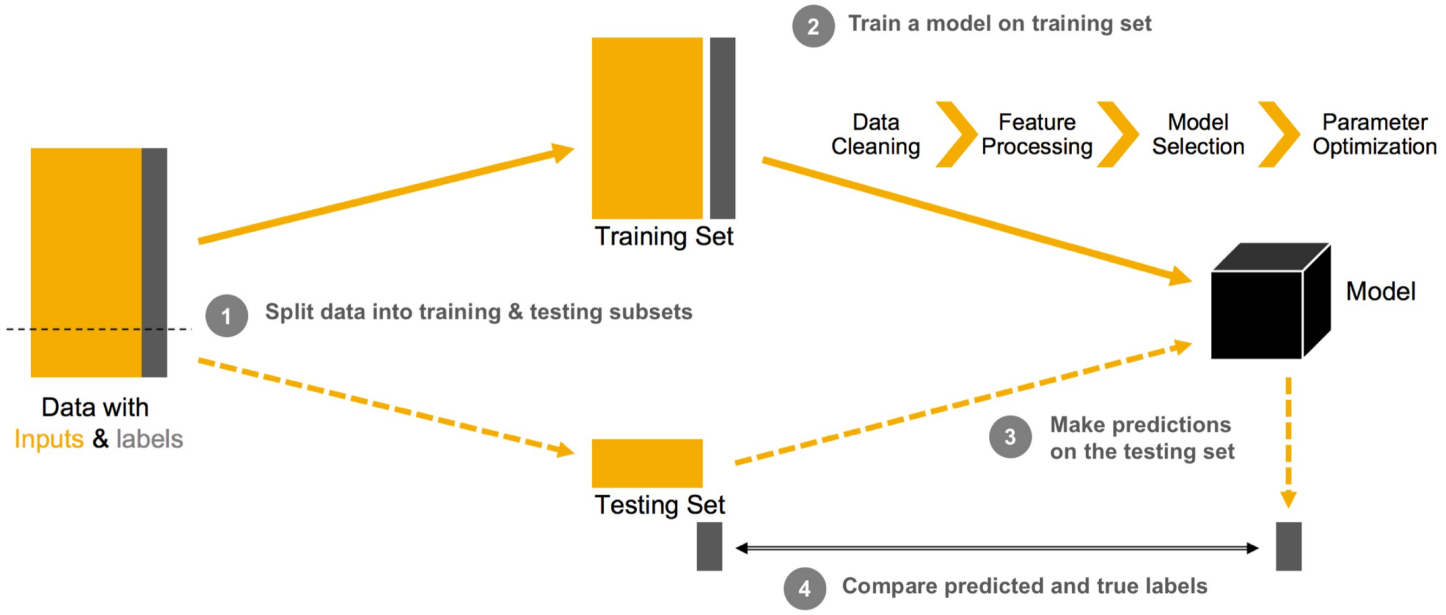
NOVA SCHOOL OF BUSINESS & ECONOMICS

The Process of Machine learning



NOVA SCHOOL OF BUSINESS & ECONOMICS

How to develop a model?



NOVA SCHOOL OF BUSINESS & ECONOMICS

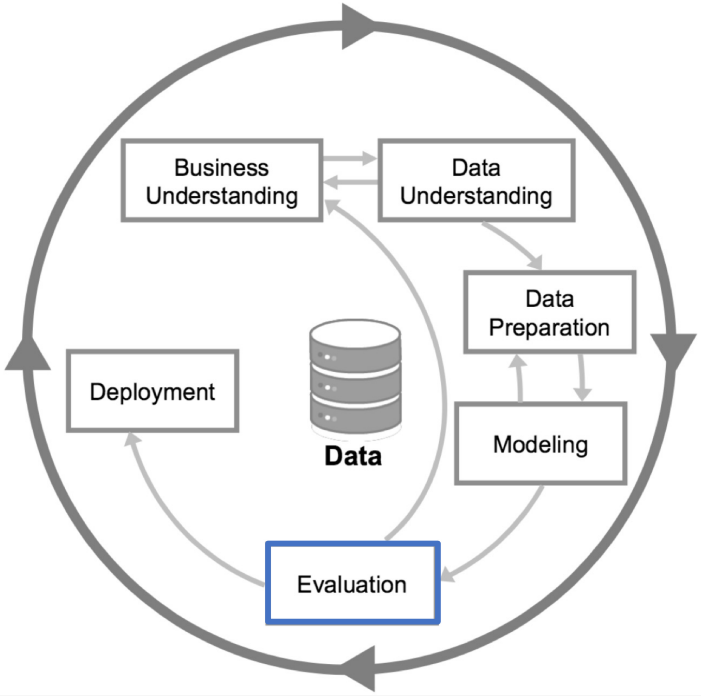
Which model should I use?

- Selecting a machine learning algorithm is mostly a process of trial and error
- **MLY #13: Build your first system quickly, then iterate**
  - Don't start off trying to design and build the perfect system. Instead, build and train a basic system quickly
- **Rule #4: Keep the first model simple and get the infrastructure right.**
  - The first model does not have to be fancy
  - Make sure you have infrastructure that does the process reliably
  - Simple model provides you with *baseline metric* that you can compare with complex models
- Accurate
  - Are we making good prediction?
- Interpretable
  - How easy is it to explain how the predictions are made?
- Fast
  - How long does it take to build a model and how long does the model take to make predictions?
- Scalable
  - How much longer do we have to wait if we build/predict using a lot of data?

https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/

NOVA SCHOOL OF BUSINESS & ECONOMICS

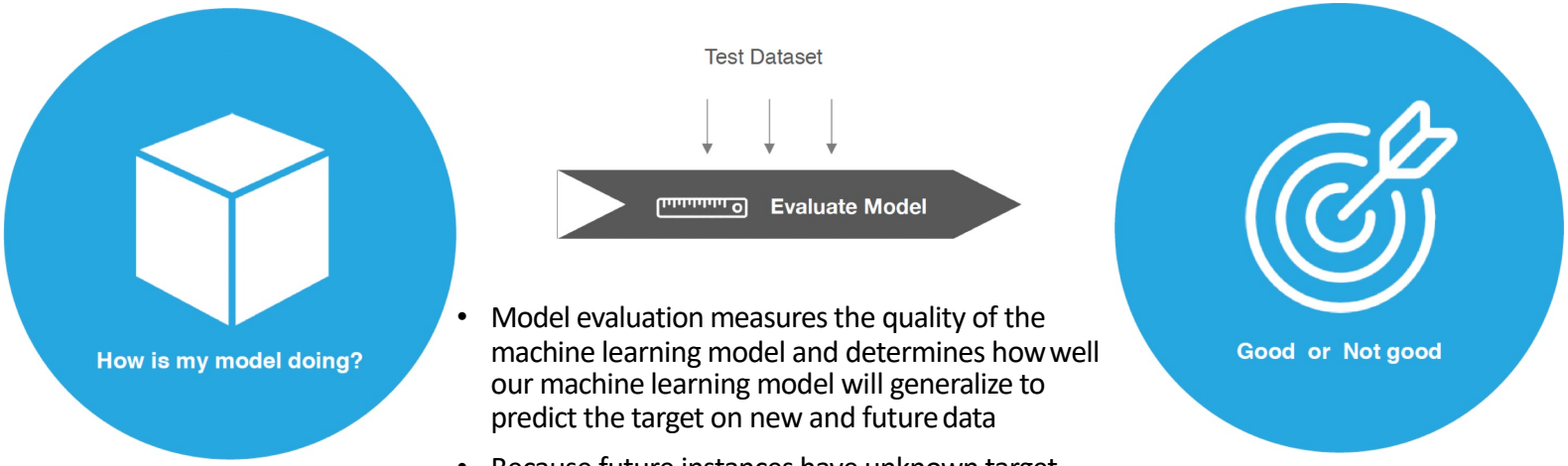
Evaluate the Model



- Evaluate performance
- Optimize parameters
- Interpret results

NOVA SCHOOL OF BUSINESS & ECONOMICS

Evaluate the model performance



- Model evaluation measures the quality of the machine learning model and determines how well our machine learning model will generalize to predict the target on new and future data
- Because future instances have unknown target values, you need to check the accuracy metric of the ML model on data for which you already know the target answer, and use this assessment as a proxy for predictive accuracy on future data

https://docs.aws.amazon.com/machine-learning/latest/dg/evaluating_models.html

NOVA SCHOOL OF BUSINESS & ECONOMICS

Model Evaluation Metrics

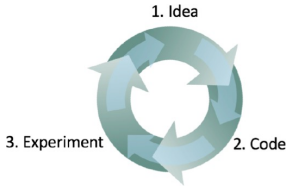
Different machine learning tasks have different performance metrics

Classification Model	Regression Model
• Accuracy	• MAE (Mean absolute error)
• Precision	• MSE (Mean Squared Error)
• Recall	• RMSE (Root means squared error)
• F score	• MAPE (Mean absolute % error)
• ROC	• R^2 (Coefficient of determination)
• AUC	
• Log Loss	

NOVA SCHOOL OF BUSINESS & ECONOMICS

Choose metrics

- **RML #13: Choose a simple, observable and attributable metric for your first objective**
  - The ML objective should be something that is easy to measure and is a proxy for the "true" objective
  - Choose a metric that is directly observed and attributed to an action of the system (e.g. is this link clicked?)
  - Avoid metric that tries to figure out hard questions (e.g. is the user happy using the product? Is the user satisfied with the recommendation?)
  - Try proxy metric instead (e.g., if the user is happy, they will stay on the site longer)
- **MLY #10: Having a validation set and metric speeds up iterations**
  1. Start off with some idea on how to build system
  2. Implement the idea in code
  3. Carry out an experiment which tells me how well the idea worked



- **MLY #11: When to change validation/test sets and metrics**
  - If you realize that your initial dev/test set or metric missed the mark, change them quickly
  - If the metric is measuring something other than what the project needs to optimize, change it quickly

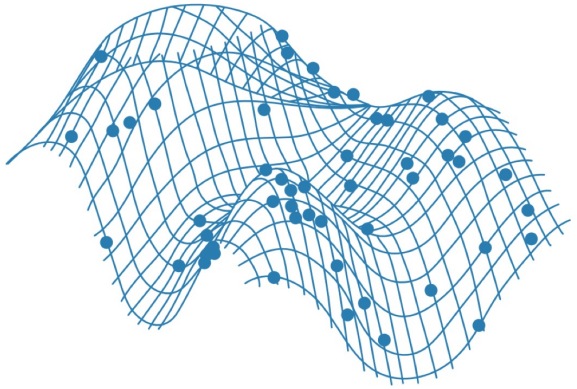
NOVA SCHOOL OF BUSINESS & ECONOMICS

Tuning the Hyperparameter

Hyper-parameter tuning is an iterative process

You begin by setting parameters based on a "best guess" of the outcome. Your goal is to find the "best possible" values—those that yield the best model. As you adjust parameters and model performance begins to improve, you see which parameter settings are effective and which still require tuning

"A simple algorithm with well-tuned parameters often produces a better model than an inadequately tuned complex algorithm."

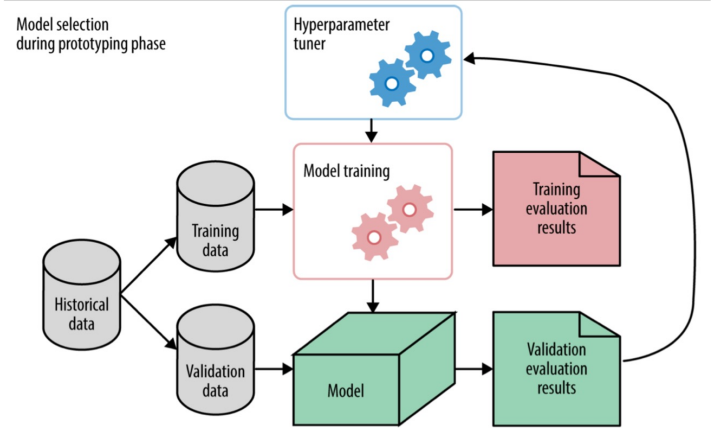


NOVA SCHOOL OF BUSINESS & ECONOMICS

Tuning the Hyperparameter

Hyperparameters tuning is the process of looking for the most optimal hyperparameters for an algorithm

Hyperparameters are specified before the training algorithm starts and can't be optimized inside the training algorithm itself. The first choice of hyperparameters will not result in optimal model performance, so hyperparameters are tuned by adjusting parameters and taking another step of training the model – essentially an optimization procedure by itself

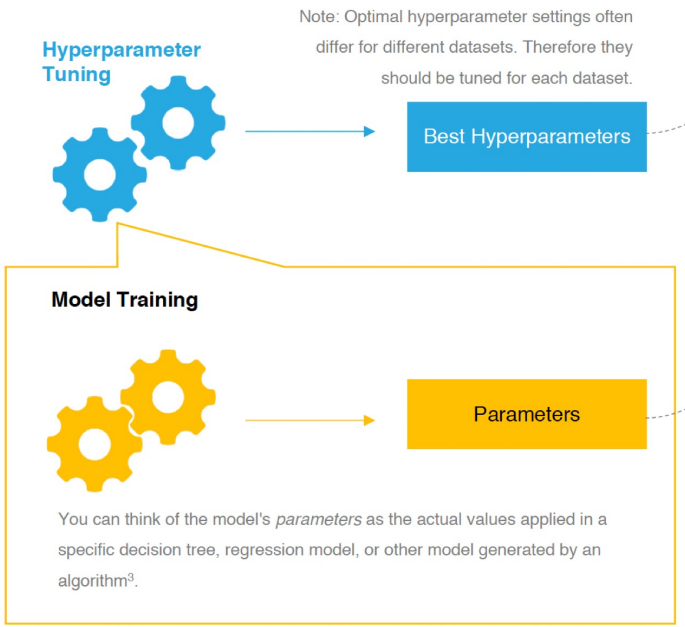


- Hyperparameter tuning is a "meta" process that controls the training process
- Hyperparameters control the model complexity
  - Regularization hyperparameters control the *capacity* of the model
- Hyperparameters control the behavior of the training algorithm
  - Optimizing a loss function during the training process also requires hyperparameters
  - Learning rate controls how fast loss function may converge

NOVA SCHOOL OF
BUSINESS & ECONOMICS

Tuning the Hyperparameter

What is the difference between Hyperparameter and Parameter?



Hyperparameters[1]

Hyperparameters are specified before the training algorithm starts and can't be optimized inside the training algorithm itself. They are external to the model.

- They are often used in processes to help estimate model parameters.
- They can not be estimated from data.
- They are often specified by the practitioner.
- They usually don't changes during a training job.

Model parameters[2]

Model's parameters are the variables that your chosen machine learning technique uses to adjust to your data. They are internal to the model.

- They are required by the model when making predictions.
- They are estimated or learned from data.
- They are often not set manually by the practitioner.
- The values are usually chosen by a learning algorithm, like the weights in linear regression or a neural network
- They changes during the a training job.

https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/tune-model-hyperparameters

NOVA SCHOOL OF BUSINESS & ECONOMICS

Tuning the Hyperparameter

The common hyperparameter optimization algorithms

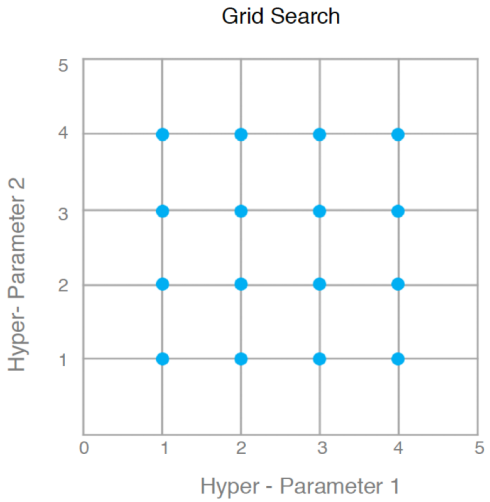
Grid Search	Randomized search	Bayesian Optimization
• A brute force search of every combination of hyperparameters	• Randomly sample and evaluate sets of hyperparameters specified by a probability distribution	• Use a statistical mode to help choose which set of hyperparameters to explore next based on the performance of past sets of hyperparameters

NOVA SCHOOL OF BUSINESS & ECONOMICS

Grid Search

Grid search tries the exhaustive searches for all the possible hyperparameter combinations

- Form a grid of hyper-parameter values
  - In practice, practitioners specify the bounds and steps between values of the hyperparameters, so that it forms a grid of configurations. we try a set of configurations of hyperparameters and train the algorithm accordingly, choosing the hyperparameter configuration that gives the best performance
- Grid search is a costly and time-consuming approach
  - This method works ok when the number of hyperparameters was relatively small. It doesn't work in cases like neural networks.



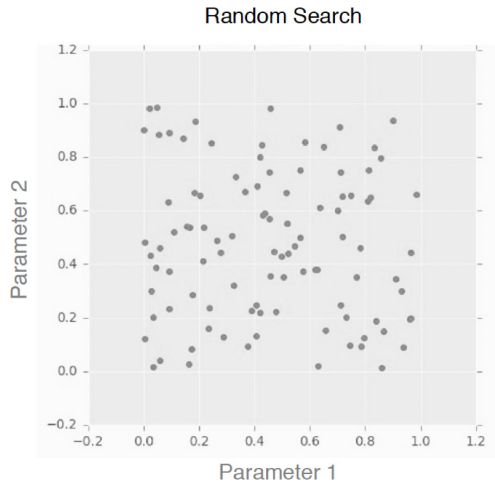
NOVA SCHOOL OF BUSINESS & ECONOMICS

Random Search

Prefer random search to grid search especially when hyperparameter search space is large

- Try random value, don't use a grid (see Andrew's explanations)
  - Instead of trying all possible combinations we will just use randomly selected subset of the hyperparameters
  - It has been found to be more effective in high-dimensional spaces than exhaustive search. This is because oftentimes, it turns out some hyperparameters do not significantly affect the loss.

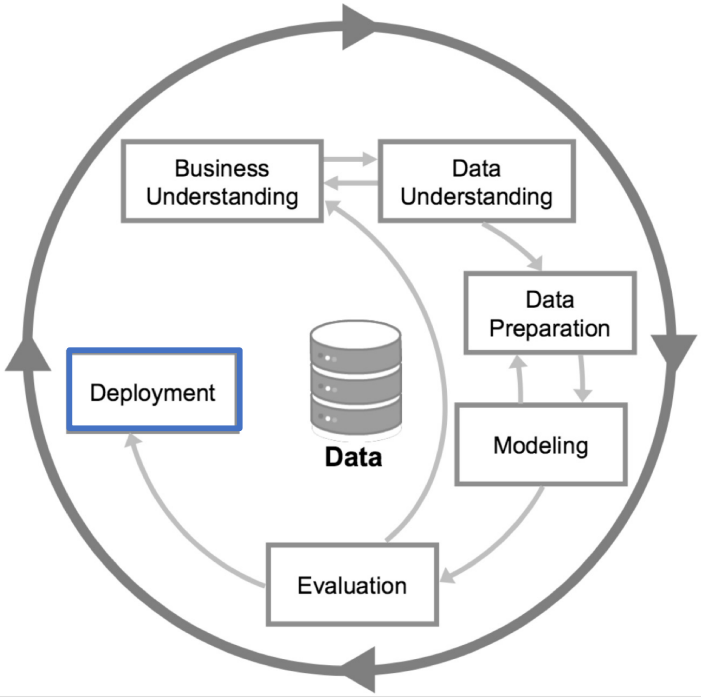
- Use a coarse to fine sampling scheme
  - In practice, it can be helpful to first search in coarse ranges (e.g. 10 ** [-6, 1]), and then depending on where the best results are turning up, narrow the range



https://www.youtube.com/watch?v=AXDByU3D1hA

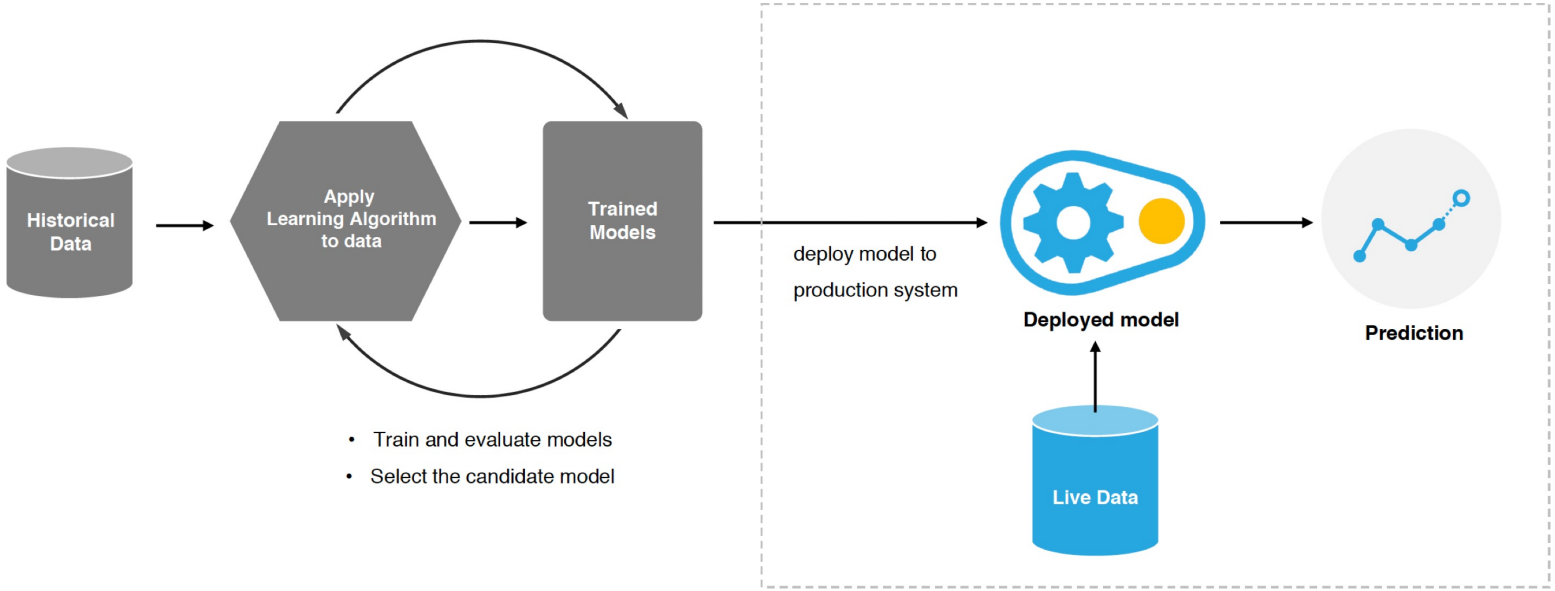
NOVA SCHOOL OF BUSINESS & ECONOMICS

Deploy the model into production



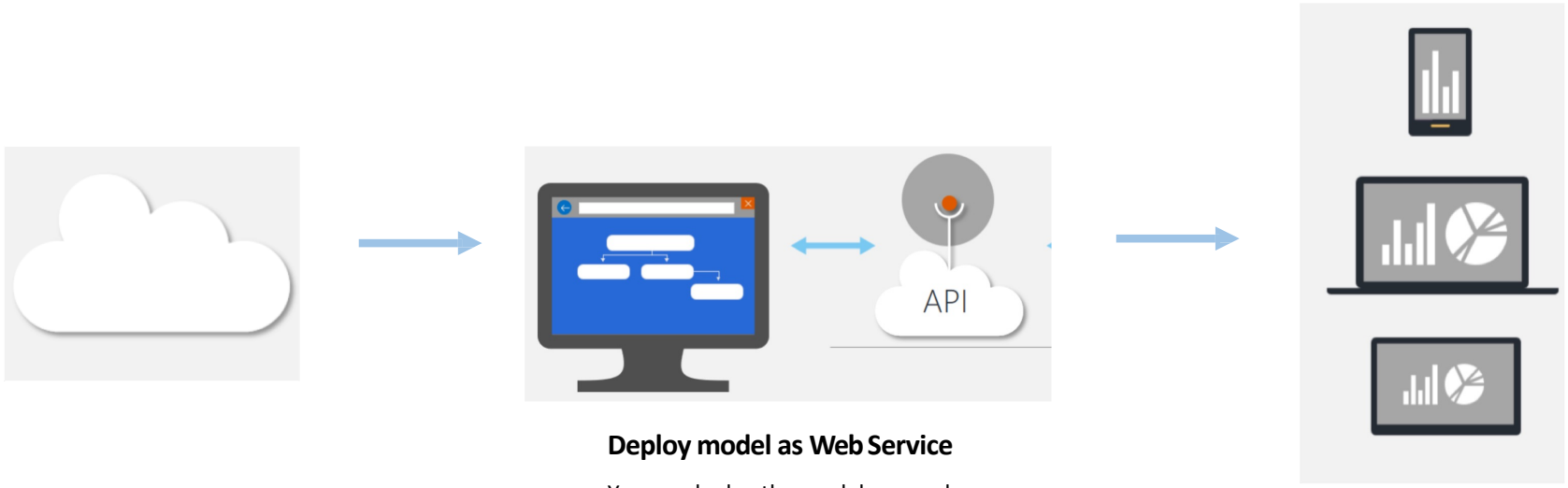
NOVA SCHOOL OF BUSINESS & ECONOMICS

Deploy phase in machine learning pipeline



NOVA SCHOOL OF BUSINESS & ECONOMICS

Deploy the trained models as web service



Deploy model as Web Service

You can deploy the model as a web service. Using the web service, users can send data to your model and the model will return its predictions

Once you deploy a Machine Learning predictive model as a Web service, you can use a REST API to send the data and get predictions

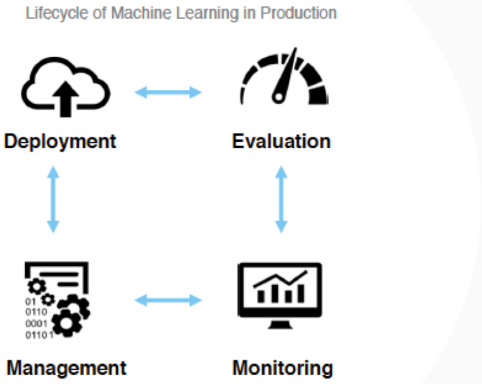
Consumer Web Service

The Web service can then be consumed in websites, business application, dashboards and mobile apps

https://docs.microsoft.com/en-us/azure/machine-learning/studio/publish-a-machine-learning-web-service

NOVA SCHOOL OF BUSINESS & ECONOMICS

What happens after (initial) deployment?



Evaluation
Measuring quality of deployed models

Monitoring
Track model quality over time ! The models have been deployed into business environment must be regularly monitored for acceptable performance. Remember - Model performance will reduce over time.

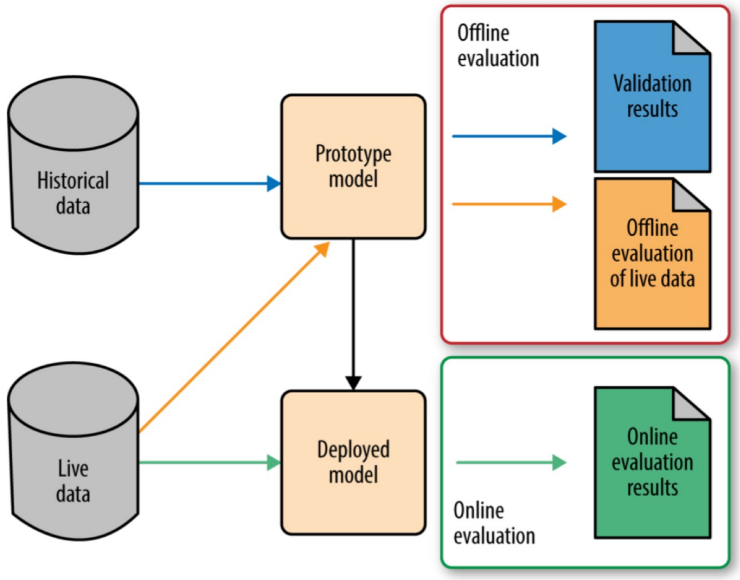
Management
Choosing between deployed models.
Improving deployed models with feedback

NOVA SCHOOL OF BUSINESS & ECONOMICS

When/how to evaluate Machine Learning Model?

Offline evaluation

- Offline evaluation measures offline metrics of the prototyped model on *historical data* (and sometimes on live data as well)
- Offline evaluation might use one of the metrics like accuracy, RMSE, etc.



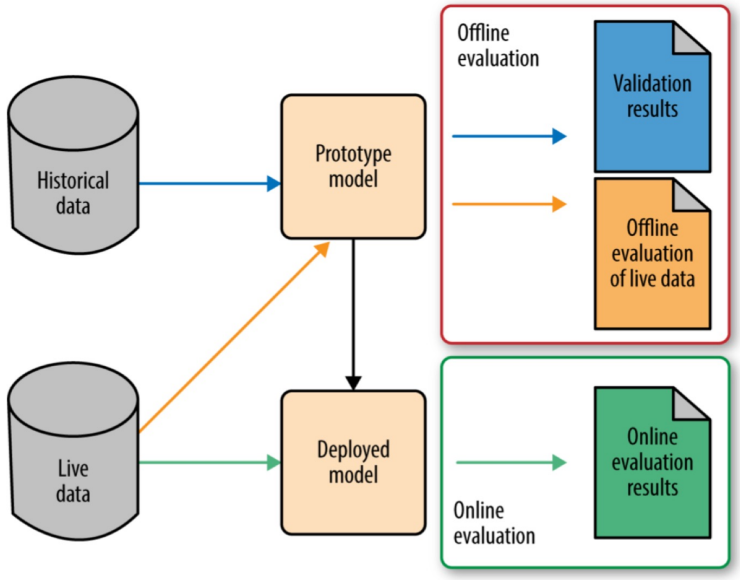
Machine learning model development and evaluation workflow

NOVA SCHOOL OF BUSINESS & ECONOMICS

When/how to evaluate Machine Learning Model?

Online evaluation

- Online evaluation measures live metrics of the deployed model on live data
- Offline metric is not equal to business metric
  - Online evaluation might measure business metrics such as Customer Lifetime Value, Click-through rate, which may not be available on historical data but are closer to what your business really cares about
- The online phase has its own testing procedure
  - The most commonly used form of online testing is A/B testing
- Track both business and ML metrics to see if they correlate



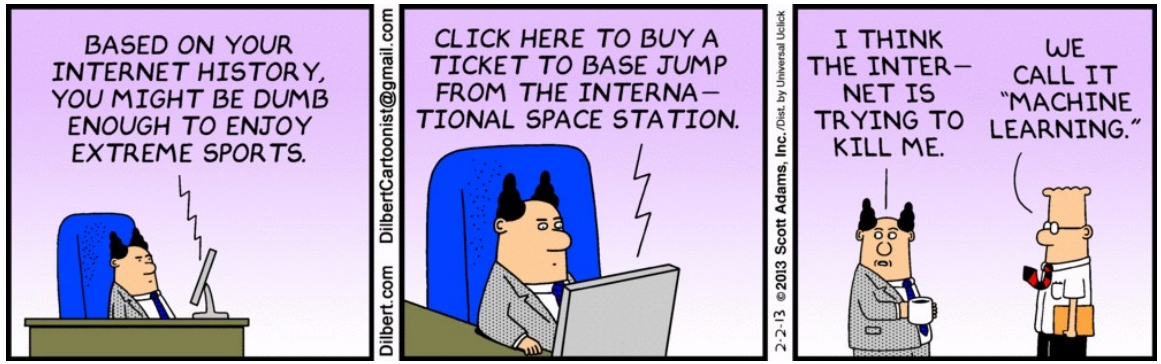
Machine learning model development and evaluation workflow

NOVA SCHOOL OF BUSINESS & ECONOMICS

Deploy into production

- **RML #5: Test the infrastructure independently from the machine learning**
  - Make sure that the infrastructure is testable, and that the learning parts of the system are encapsulated so that you can test everything around it
  - Test getting data into the algorithm. Check statistics in your pipeline in comparison to statistics for the same data processed elsewhere
  - Test getting models out of the training algorithm. Make sure that the model in your training environment gives the same score as the model in your serving environment
- **MLY #8: Know the freshness requirements of your system**
  - How much does performance degrade if you have a model that is a day old? A week old? A quarter old?
  - If you lose significant product quality if the model is not updated for a day, it makes sense to have an engineer watching it continuously
- **MLY #9: Detect problems before exporting models**
  - Many machine learning systems have a stage where you export the model to serving
  - Specifically, make sure that the model's performance is reasonable on held out data before deployment

NOVA SCHOOL OF BUSINESS & ECONOMICS



Questions?

NOVA SCHOOL OF BUSINESS & ECONOMICS