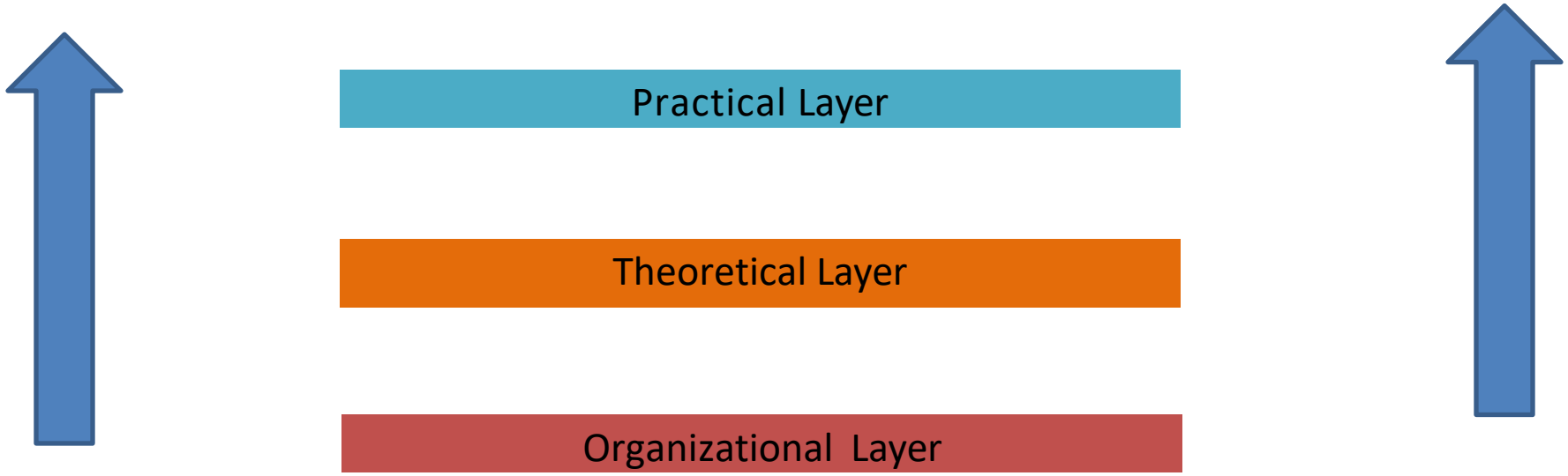
Processing Big Data For Analytics (Day 1 and Day 2 AM)

Stuart Wang

c.wang@nus.edu.sg

Agenda

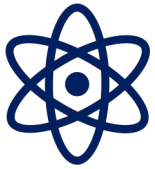


Contents

Reason for Data/AI Projects Failure

Design a Strategy to Create a Business Value

Workflow Design



Data Pipeline (Possibly Day 2 AM)

Tools and Workshop (Day 2 AM)

Reason for Data/AI Projects Failure

Do you know that…

- Only one in two organizations has moved beyond pilots and proofs of concept according to CapGemini Research Institute in 2020.
- Moreover, 72% of a cohort of organizations that began AI pilots before 2019 have not been able to deploy even a single application in production one year later.
- Algorithmia's survey in 2020 found that 55% of companies surveyed have not deployed an ML model.
- If you search with key words like "failing big data projects", you would get many results.

- To summarize, it looks like models don't make it into production and if they do, they break.

- But why?

Food for Thought

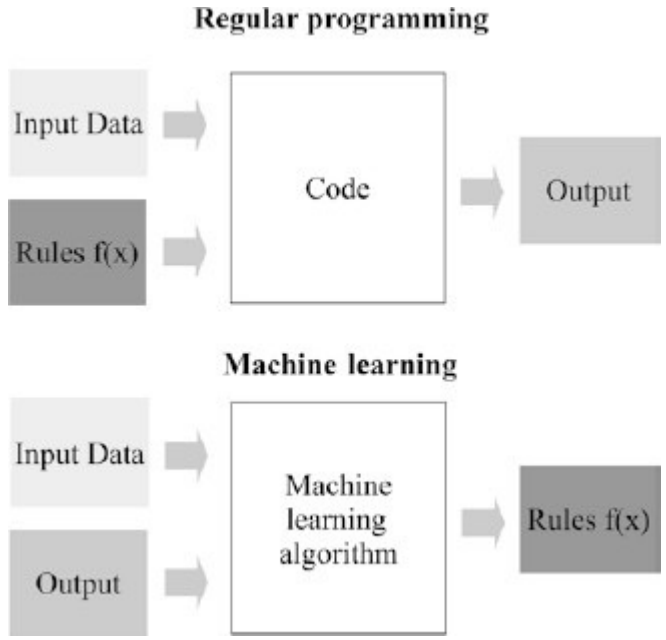
Why is that so?

We can investigate this from many different perspective.

Due to time constraint, today we are going to investigate from the knowledge gap perspective as well as the hidden technical debt in machine learning systems perspective. Many of those debts/knowledge gaps cause AI/Big Data projects to fail.

Knowledge Gap

- Knowledge Gaps
  - The Data Scientist Knowledge Gap
  - IT Knowledge Gap
  - Technology Knowledge Gap
  - Leadership Knowledge Gap
  - Data Literacy Gap
- Outdated Approaches to Managing Data and Product Analytics
- Lack of Support for Data analytics



Ref: Atwal, Harvinder "Practical Data Ops" , O'Reilly 2019 Print

Food for Thought

"Create me a machine translation attention model using a bi-directional long short-term memory (LSTM) with an attention layer outputting to a stacked post-attention LSTM feeding a softmax layer for predictions," said no CEO, ever.

Entanglement

- Machine learning system mix signals together, entangling them and making isolation of improvements become impossible.
  - CACE principle: Changing Anything Changes Everything.
  - CACE applies not only to input signals, but also to hyper-parameters, learning settings, sampling methods, convergence thresholds, data selections and essentially every other possible tweak.
- One possible mitigation strategy is to isolate models and serve ensembles.

Undeclared Consumers

- Oftentimes, a prediction from a machine learning model m_a is made widely accessible, either at runtime or by writing to files or logs that may later be consumed by other systems.

- Without access controls, some of these consumers may be *undeclared*, silently using the output of a given model as an input to another system. In more classical software engineering, these issues are referred to as visibility debt.

- Undeclared consumers are expensive at best and dangerous at worst, because they create a hidden tight coupling of model m_a to other parts of the stack. Changes to m_a will very likely impact these other parts, potentially in ways that are unintended, poorly understood, and detrimental.

Unstable Data Dependencies

- To move quickly, it is often convenient to consume signals as input features that are produced by other systems.
- However, some input signals are *unstable*, meaning that they qualitatively or quantitatively change behavior over time.
- For example, consider the case in which an input signal was previously mis-calibrated. The model consuming it likely fit to these mis-calibrations, and a silent update that corrects the signal will have sudden ramifications for the model.
- One common mitigation strategy for unstable data dependencies is to create a *versioned copy* of a given signal.

Glue Code

- ML researchers tend to develop general purpose solutions as self-contained packages.
- Using generic packages often results in a glue code system design pattern, in which a massive amount of supporting code is written to get data into and out of general-purpose packages.
- Glue code is costly in the long term because it tends to freeze a system to the peculiarities of a specific package; testing alternatives may become prohibitively expensive.
- In this way, using a generic package can inhibit improvements, because it makes it harder to take advantage of domain-specific properties or to tweak the objective function to achieve a domain-specific goal.
- Therefore, native clean solution is quite relevant and might save cost in the long run.

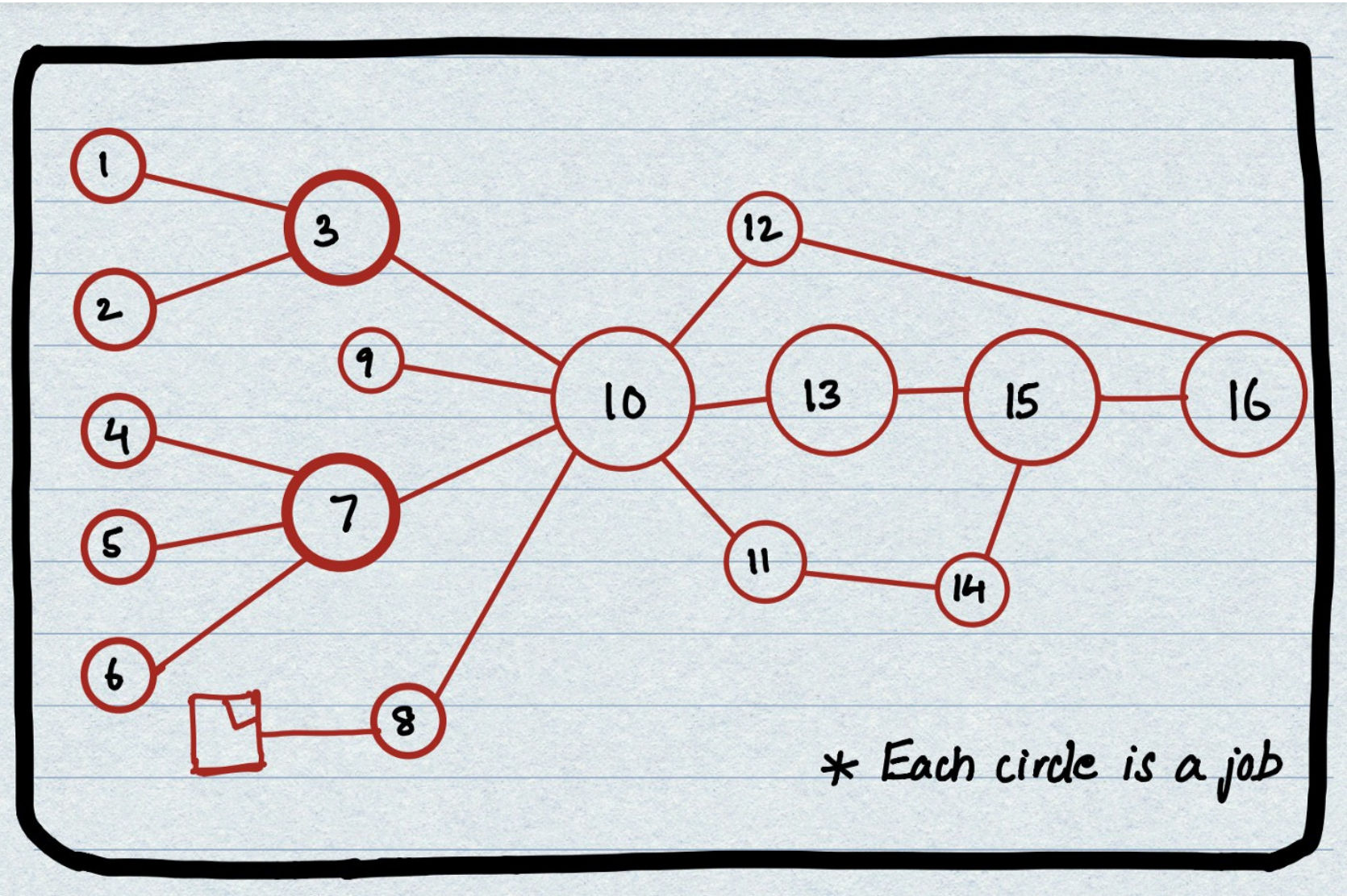
Pipeline Jungles

- As a special case of glue code, *pipeline jungles* often appear in data preparation. These can evolve organically, as new signals are identified and new information sources added incrementally.

- Without care, the resulting system for preparing data in an ML-friendly format may become a jungle of scrapes, joins, and sampling steps, often with intermediate files output.

- Pipeline jungles can only be avoided by thinking holistically about data collection and feature extraction.

How Pipeline Jungle Looks Like



Culture

- There is sometimes a hard line between ML research and engineering, but this can be counter-productive for long-term system health.

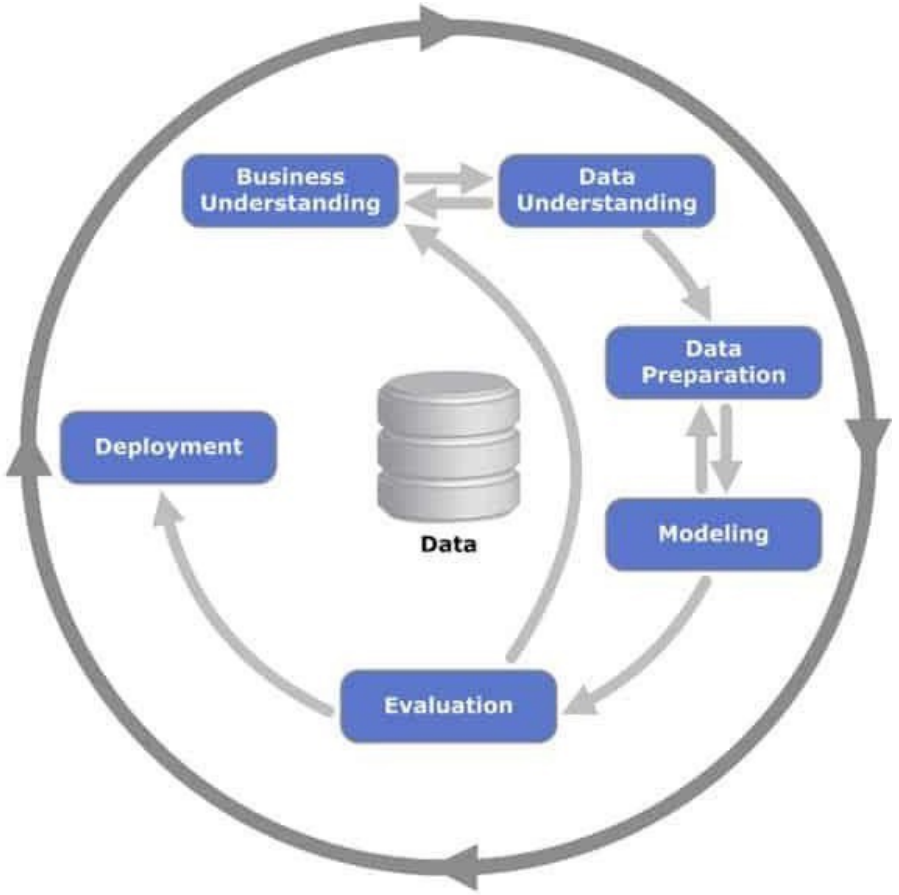
- It is important to create team cultures that reward deletion of features, reduction of complexity, improvements in reproducibility, stability, and monitoring to the same degree that improvements in accuracy are valued.

Design a Strategy to create a business Value

- CRISP-ML
- MicrosoI TDSP
- Value Proposition Canvas
- Business Model Canvas
- AI Canvas
- Machine Learning Canvas

CRISP - ML

CRISP DM



CRISP-DM is Still the Most Popular Framework for Executing Data Science Projects

Business Understanding

Determine Business Objectives
Background
Business Objectives
Business Success Criteria

Assess Situation
Inventory of Resources
Requirements, Assumptions, and Constraints
Risks and Contingencies
Terminology
Costs and Benefits

Determine Data Mining Goals
Data Mining Goals
Data Mining Success Criteria

Produce Project Plan
Project Plan
Initial Assessment of Tools and Techniques

Data Understanding

Collect Initial Data
Initial Data Collection Report

Describe Data
Data Description Report

Explore Data
Data Exploration Report

Verify Data Quality
Data Quality Report

Data Preparation

Select Data
Rationale for Inclusion/ Exclusion

Clean Data
Data Cleaning Report

Construct Data
Derived Attributes
Generated Records

Integrate Data
Merged Data

Format Data
Reformatted Data

Dataset
Dataset Description

Modeling

Select Modeling Techniques
Modeling Technique
Modeling Assumptions

Generate Test Design
Test Design

Build Model
Parameter Settings
Models
Model Descriptions

Assess Model
Model Assessment
Revised Parameter Settings

Evaluation

Evaluate Results
Assessment of Data Mining Results w.r.t. Business Success Criteria
Approved Models

Review Process
Review of Process

Determine Next Steps
List of Possible Actions
Decision

Deployment

Plan Deployment
Deployment Plan

Plan Monitoring and Maintenance
Monitoring and Maintenance Plan

Produce Final Report
Final Report
Final Presentation

Review Project
Experience Documentation

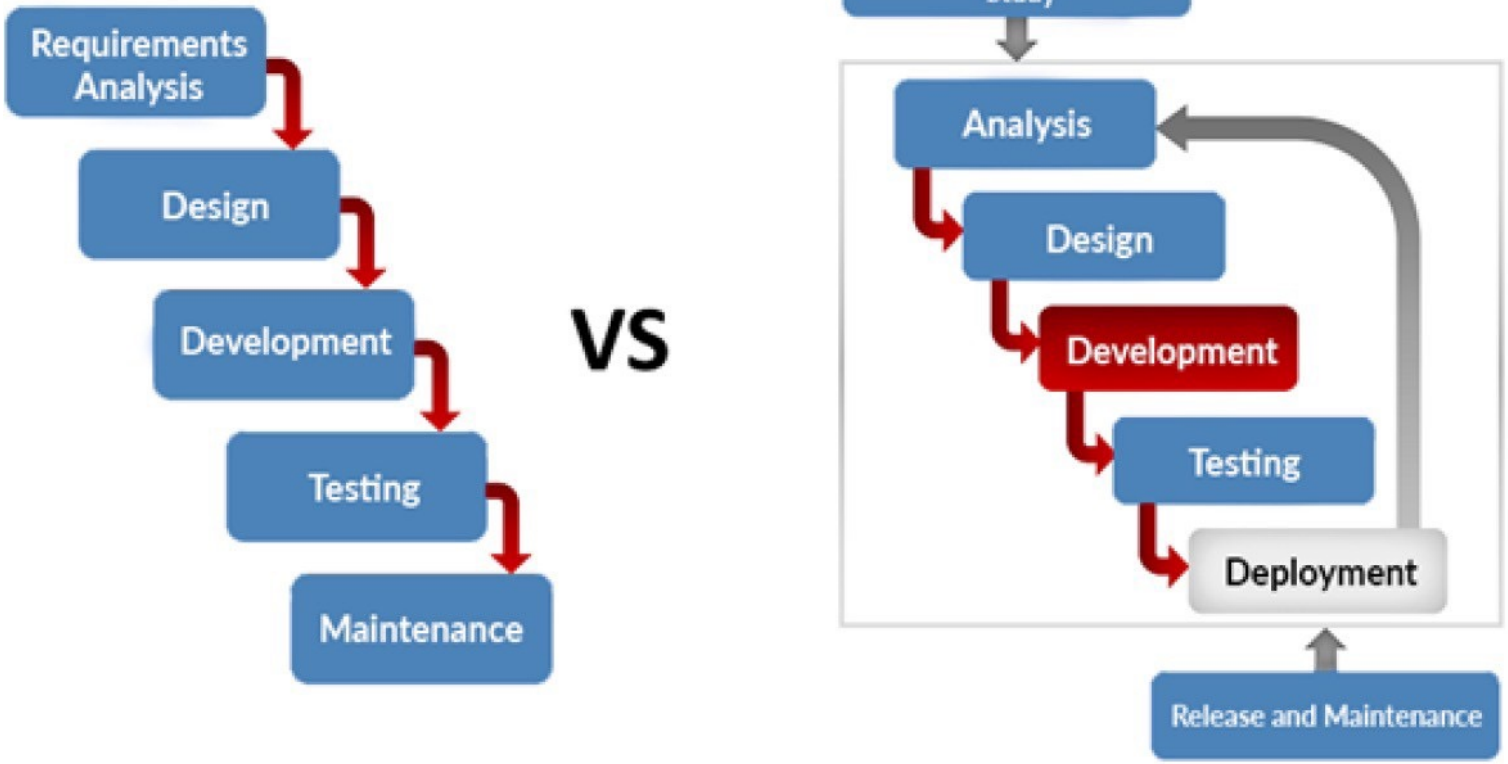
CRISP-DM

- CRISP-DM activities are organized in six phases. The successful completion of a phase initiates the execution of the subsequent activity.

- CRISP-DM includes iterations of revisiting previous steps until success or completion criteria are met. It can be therefore characterized as a waterfall life cycle with backtracking.

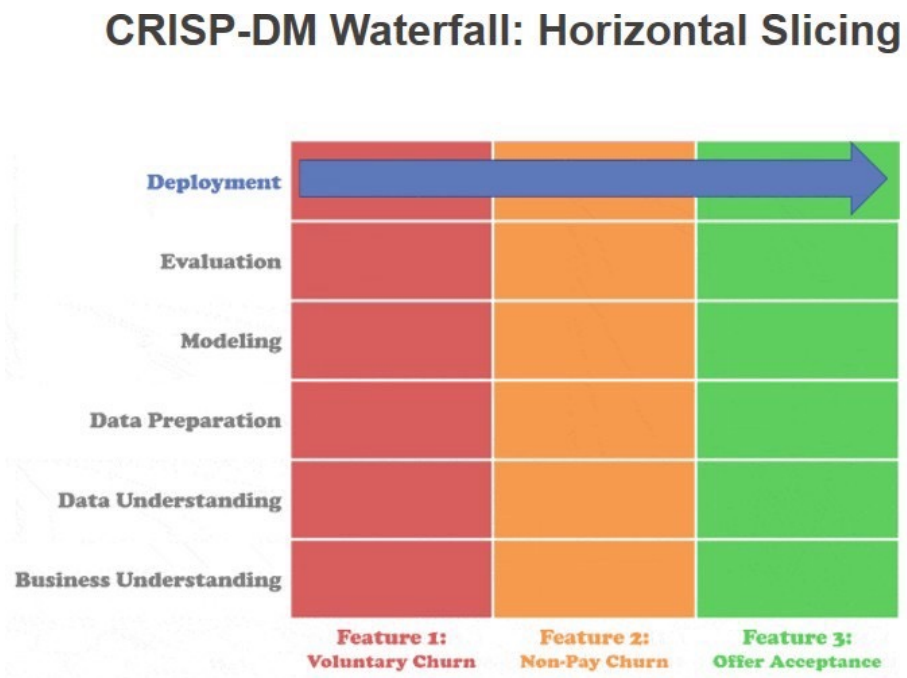
- Generally, if you follow CRISP-DM in a more flexible way, iterate quickly, you will wind up with an agile approach. If you follow CRISP-DM precisely and choose not to iterate frequently, you are operating a waterfall process. You take someway in the middle, that is waterfall life cycle with backtracking as mentioned above.

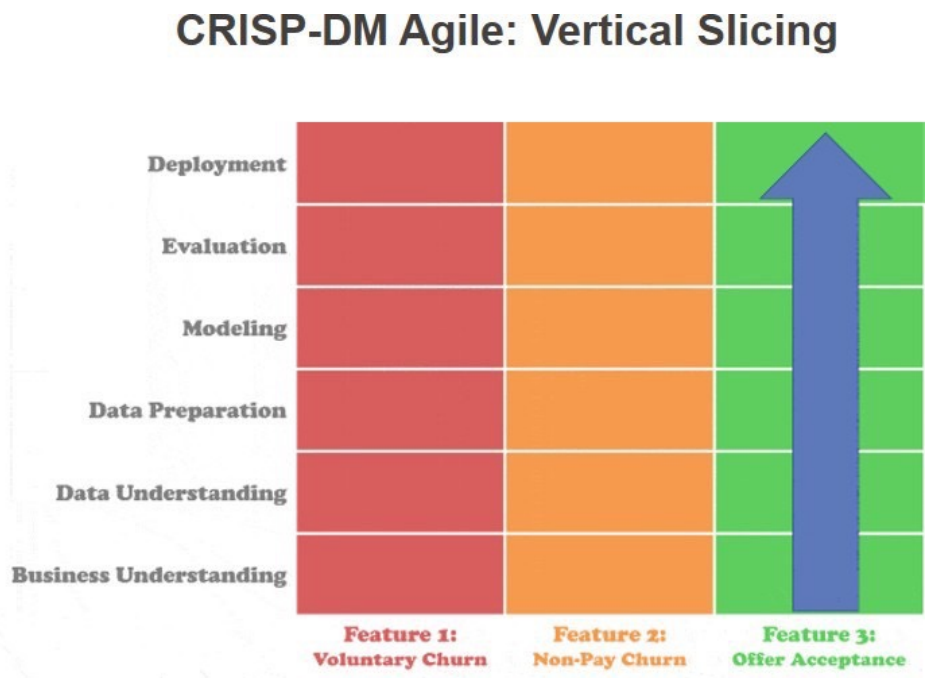
Waterfall & Agile



Ref: Emmanuel Raj “Engineering ML Ops” , Packt 2021 Print

CRISP DM (Agile or Waterfall?)





Shortcomings of CRISP-DM

- CRISP-DM focuses on data mining and does not cover the application scenario of ML models inferring real-time decisions over a long period of time.
  - The ML model has to be adaptable to a changing environment or the model's performance will degrade over time, such that, a permanent monitoring and maintaining of the ML model is required after the deployment

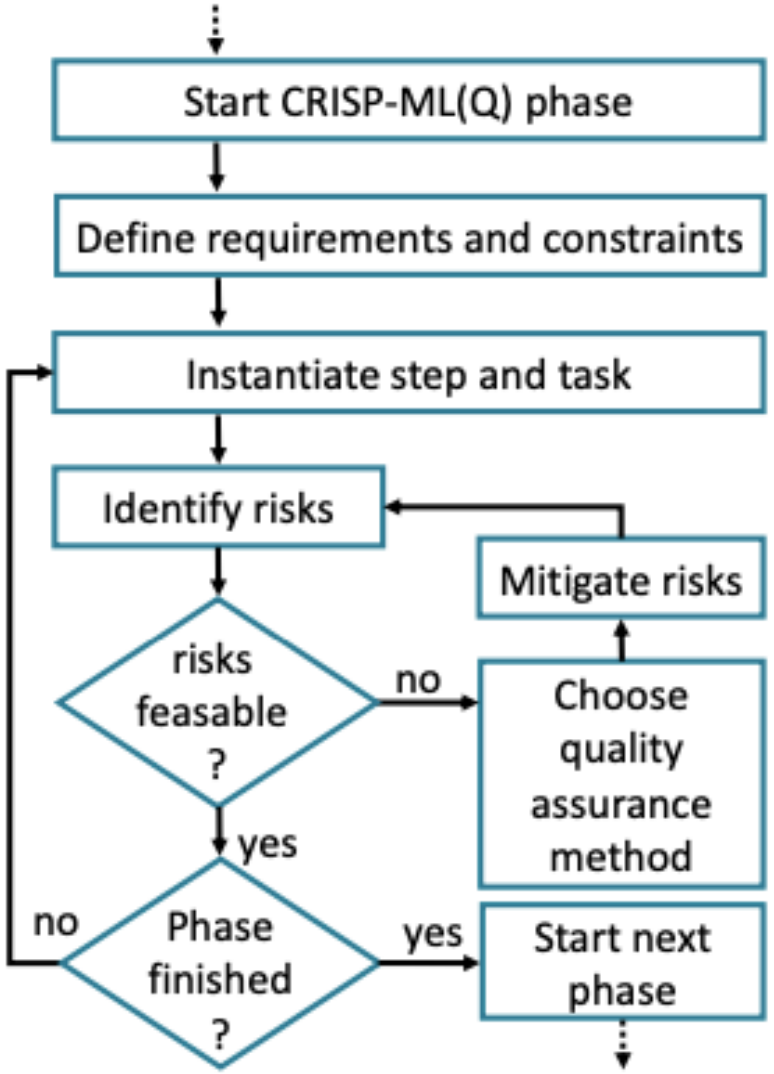
- More importantly, CRISP-DM lacks guidance on quality assurance methodology.
  - Quality is not only defined by the product's fitness for its purpose, but the quality of the task executions in any phase during the development of a ML application.

Introducing CRISP-ML (Q)

- Cross Industry Standard Process model for the development of Machine Learning applications with Quality assurance methodology.
- As a first contribution, quality assurance methodology is introduced in each phase and tasks of the process model.
  - The quality methodology serves to mitigate risks that affect the success and efficiency of the machine learning application
- As a second contribution, CRISP-ML(Q) covers a monitoring and maintenance phase to address risks of model degradation in a changing environment.
- Moreover, business and data understanding are merged into a single phase because industry practice has taught us that these two activities, which are separate in CRISP-DM, are strongly intertwined.

Studer, S.; Bui, T.B.; Drescher, C.; Hanuschkin, A.; Winkler, L.; Peters, S.; Müller, K.-R.; Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Preprints* **2021**, *1*, 0. https:titidoi.orgti

CRISP-ML(Q)	CRISP-DM
Business & Data Understanding	Business Understanding
	Data Understanding
Data Preparation	Data Preparation
Modeling	Modeling
Evaluation	Evaluation
Deployment	Deployment
Monitoring & Maintenance	-



red highlights -- data related
blue highlights -- model related

Studer, S.; Bui, T.B.; Drescher, C.; Hanuschkin, A.; Winkler, L.; Peters, S.; Müller, K.-R.; Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Preprints* **2021**, *1*, 0.

Business and Data Understanding

- Developing machine learning applications starts with identifying the scope of the ML application, the success criteria, and a data quality verification. The goal of this first phase is to ensure the feasibility of the project.

- Confirming the feasibility before setting up the ML project is a best practice in an industrial setting. Applying the Machine Learning Canvas framework would be a structured way to perform this task.

- The ML Canvas guides through the prediction and learning phases of the ML application. In addition, it enables all stakeholders to specify data availability, regulatory constraints, and application requirements such as robustness, scalability, explainability, and resource demand.

Feasibility Check

- Checking feasibility before setting up the project is considered best practice for the overall success of the ML approach and can minimize the risk of premature failures due to false expectations.
  - Data: in practice, a major source of project delay is the lack of data availability. A small sample size carries the risk of low performance on out-of-sample data. The risk might be mitigated by e.g. adding domain knowledge or increasing data quality. However, if the sample size is not sufficient the ML project should be terminated or put on hold at this stage.
  - Applicability of ML technology: it is common to demonstrate the feasibility of a ML application with a proof of concept (PoC) when the ML algorithm is used for the first time in a specific domain. If PoC already exists, setting up a software project that focuses on the deployment directly is more sufficient.
  - Legal constraints: legal constraints are frequently augmented by ethical and social considerations like fairness and trust.
  - Requirements on the application: requirements could include robustness, scalability, explainability and resources demand. The challenge during the development is to optimize the success criteria while not violating the requirements and constraints.

Success Criteria

- Business Success Criteria
  - For example, if an ML application is planned for a quality check in production and is supposed to outperform the current manual failure rate of 3%, the business success criterion could be derived as e.g. “failure rate less than 3%”
- ML Success Criteria (translate the business objectives into ML success criteria)
  - using the example above, the ML success criteria is defined as “accuracy greater than 97%”.
- Economic Success Criteria
  - It is best practice to add an economic success criterion in the form of a Key Performance Indicator (KPI) to the project.
  - A KPI is an economical measure for the relevance of the ML application.
  - Using the same example, KPI can be defined as “cost savings with automated quality check per part”.

Data Preparation

- The second phase of the CRISP-ML(Q) process model aims to prepare data for the following modeling phase. *Data selection, data cleaning, feature engineering,* and *data standardization* tasks are performed during this phase.

- We identify valuable and necessary features for future model training by using either filter methods, wrapper methods, or embedded methods for data selection.

- For example, at this stage, you could
  - discard sample that do not satisfy data quality requirements
  - tackle the problem of unbalanced classes by applying over-sampling or under-sampling strategies
  - identify valuable and necessary features for future model training via feature engineering

- We will cover more details for examples above in later slides

Data Preparation
Data Collection

- Costs and time is needed to collect enough consistent data by preparing and merging data from different sources and different formats.

- A ML project might be delayed until the data is collected or could even be stopped if the collection of data of sufficient quality is not feasible.

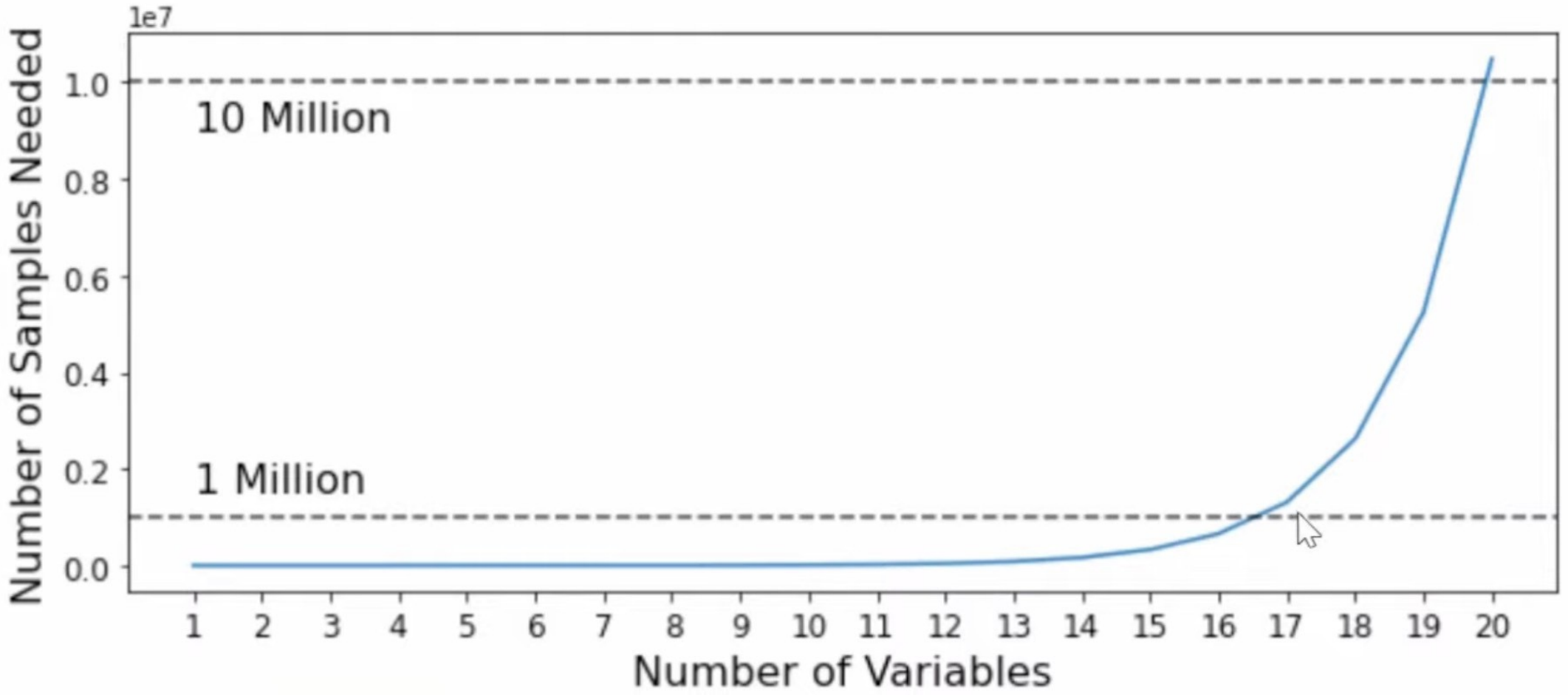
- Data Version Control
  - Collectting data is not a static task but rather an iterative task.
  - Modification on the data set should be documented to mitigate the risk of obtaining irreproducible or wrong results.
  - Version control on the data is one of the essential tools to assure reproducibility and quality as it allows to track errors and unfavorable modifications during the development.

Data Preparation
Data Selection

- Feature selection
  - Intuitively an exponentially increasing number of samples from an increasing number of features is required to prevent the data from becoming sparse in the feature space. This is termed as the curse of dimensionality.
  - Therefore, it is best practice to select just necessary features.
  - The selection of features should not be relied purely on the validation and test error but should be analyzed by a domain expert as potential biases might occur due to spurious correlations.

- Data Selection
  - Discarding samples should be well documented and strictly based on objective quality criteria.

- Unbalanced Classes
  - Over-sampling of the minority and/or under-sampling of the majority class could improve the situation where the number of samples per class is skewed.



Modelling

- The modeling phase is the ML-specific part of the process. This phase aims to specify one or several machine learning models to be deployed in the production.

- The translation to the ML task depends on the business problem that we are trying to solve. Constraints and requirements from the Business and Data Understanding phase will shape this phase.

- For example, the application domain's model assessment metrics might include performance metrics, robustness, fairness, scalability, interpretability, model complexity degree, and model resource demand. We should adjust the importance of each of these metrics according to the use case.

Evaluation

- Consequently, model training is followed by a model evaluation phase, also known as offline testing. During this phase, the performance of the trained model needs to be validated on a test set. Additionally, the model robustness should be assessed using noisy or wrong input data.

- Furthermore, it is best practice to develop an explainable ML model to provide trust, meet regulatory requirements, and govern humans in ML-assisted decisions.

- Finally, the model deployment decision should be met automatically based on success criteria or manually by domain and ML experts. Similar to the modeling phase, all outcomes of the evaluation phase need to be documented.

Quality Measure of Machine Learning Models

Performance	The model's performance on unseen data.
Robustness	The model's resiliency to inconsistent inputs and to failures in the execution environment.
Scalability	The model's ability to scale to high data volume in the production system.
Explainability	The model's direct or post-hoc explainability.
Model Complexity	The model's capacity should suit the data complexity.
Resource Demand	The model's resource demand for deployment.

Studer, S.; Bui, T.B.; Drescher, C.; Hanuschkin, A.; Winkler, L.; Peters, S.; Müller, K.-R.; Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Preprints* **2021**, *1*, 0. https:titidoi.org
ti

Deployment

- The ML model deployment includes the following tasks:
  - *hardware definition*
  - *model evaluation* in a production environment (online testing, e.g., A/B tests)
  - providing *user acceptance and usability tes+ng*
  - providing a *fall-back plan for model outages*
  - setting up the *deployment strategy* to roll out the new model gradually (e.g. canary or green/blue deployment).

Monitoring and Maintenance

- Once the ML model has been put into production, it is essential to monitor its performance and maintain it.
  - When an ML model performs on real-world data, the main risk is the “model staleness” effect when the performance of the ML model drops as it starts operating on unseen data.
  - Furthermore, model performance is affected by hardware performance and the existing software stack.
  - Therefore, the best practice to prevent the model performance drop is to perform the monitoring task when the model performance is continuously evaluated to decide whether the model needs to be re-trained. This is known as the Continued Model Evaluation.
  - The decision from the monitoring task leads to the second task - updating the ML model.
  - Additionally to monitoring and re-training, reflecting on the business use case and the ML task might be valuable for adjusting the ML process.

Monitoring and Maintenance

- Non-stationary data distribution
  - Data distributions change over time and the characteristics of the data distribution are represented incorrectly by the stale training data set. This degrades the performance of the model over time.
- Degradation of hardware
  - The hardware that the model is deployed on and the sensor hardware will age over time. Sensors get noisier or fail over time. This will shift the domain of the system and has to be adapted by the model or by retraining it.
- System updates
  - Updates on the software or hardware of the system can cause a shift in the environment.

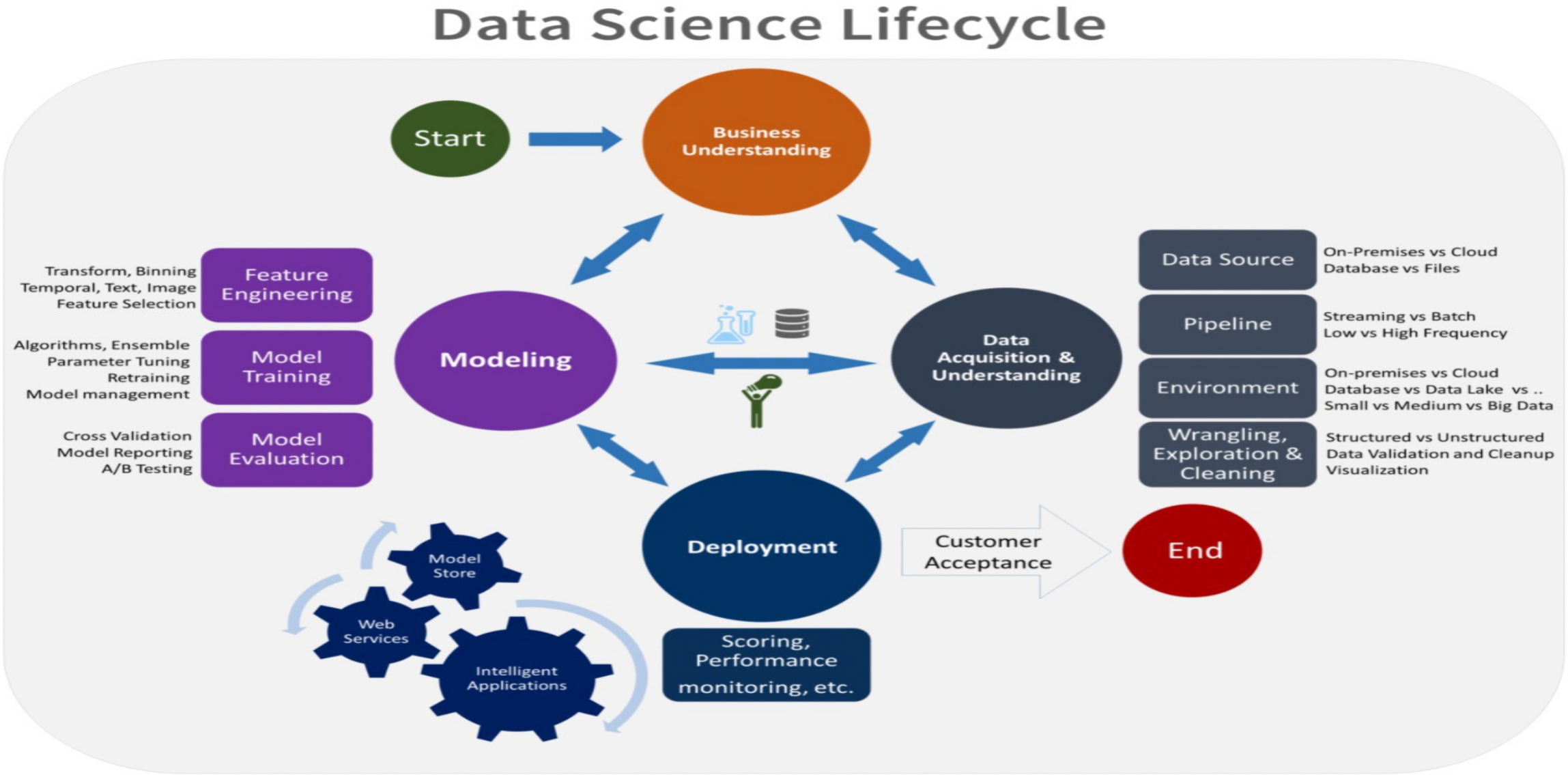
Microsoft TDSP

Data Mining Life Cycles: Stemming from the 1990s are the traditional data mining life cycles such as CRISP-DM that tend to focus narrowly on the core data and modeling aspects of the project life cycle.

Modern Data Science Life Cycles: More recently, organizations have started to define their own frameworks that are more modern takes to the general project life cycle in that they:

- integrate agile principles and practices
- acknowledge the multiple team roles of a data science project
- extend the core data and modeling phases to first focus on the business problem and to finish with deployment

Microsoft TDSP



Business Understanding

Goals

Specify the key variables that are to serve as the model targets and whose related metrics are used determine the success of the project.

Identify the relevant data sources that the business has access to or needs to obtain.

How to do it

There are two main tasks addressed in this stage:

- **Define objectives**: Work with your customer and other stakeholders to understand and identify the business problems. Formulate questions that define the business goals that the data science techniques can target.
- **Identify data sources**: Find the relevant data that helps you answer the questions that define the objectives of the project.

Artifacts

Here are the deliverables in this stage:

- **Charter document**: A standard template is provided in the TDSP project structure definition. The charter document is a living document. You update the template throughout the project as you make new discoveries and as business requirements change. The key is to iterate upon this document, adding more detail, as you progress through the discovery process. Keep the customer and other stakeholders involved in making the changes and clearly communicate the reasons for the changes to them.

- **Data sources**: This section specifies the original and destination locations for the raw data. In later stages, you fill in additional details like the scripts to move the data to your analytic environment.

- **Data dictionaries**: This document provides descriptions of the data that's provided by the client. These descriptions include information about the schema (the data types and information on the validation rules, if any) and the entity-relation diagrams, if available.

Data Acquisition and Understanding

Goals

- Produce a clean, high-quality data set whose relationship to the target variables is understood. Locate the data set in the appropriate analytics environment so you are ready to model.
- Develop a solution architecture of the data pipeline that refreshes and scores the data regularly.

How to do it

There are three main tasks addressed in this stage:

- **Ingest the data** into the target analytic environment.
- **Explore the data** to determine if the data quality is adequate to answer the question.
- **Set up a data pipeline** to score new or regularly refreshed data.

Artifacts

The following are the deliverables in this stage:

- **Data quality report**: This report includes data summaries, the relationships between each attribute and target, variable ranking, and more.
- **Solution architecture**: The solution architecture can be a diagram or description of your data pipeline that you use to run scoring or predictions on new data after you have built a model. It also contains the pipeline to retrain your model based on new data.
- **Checkpoint decision**: Before you begin full-feature engineering and model building, you can re-evaluate the project to determine whether the value expected is sufficient to continue pursuing it. You might, for example, be ready to proceed, need to collect more data, or abandon the project as the data does not exist to answer the question.

Modelling

Goals

- Determine the optimal data features for the machine-learning model.
- Create an informative machine-learning model that predicts the target most accurately.
- Create a machine-learning model that's suitable for production.

How to do it

There are three main tasks addressed in this stage:

- **Feature engineering**: Create data features from the raw data to facilitate model training.
- **Model training**: Find the model that answers the question most accurately by comparing their success metrics.
- Determine if your model is **suitable for production.**

Machine Learning Algorithm Cheat Sheet

This cheat sheet helps you choose the best machine learning algorithm for your predictive analytics solution.
Your decision is driven by both the nature of your data and the goal you want to achieve with your data.

Microsoft Azure

What do you want to do?

Extract information from text

Text Analytics

Derives high-quality information from text

Answers questions like: What info is in this text?

- **Latent Dirichlet Allocation** ← Unsupervised topic modeling, group texts that are similar
- **Extract N-Gram Features from Text** ← Creates a dictionary of n-grams from a column of free text
- **Feature Hashing** ← Converts text data to integer encoded features using the Vowpal Wabbit library
- **Preprocess Text** ← Performs cleaning operations on text, like removal of stop-words, case normalization
- **Word2Vector** ← Converts words to values for use in NLP tasks, like recommender, named entity recognition, machine translation

Predict between several categories

Multiclass Classification

Answers complex questions with multiple possible answers

Answers questions like: Is this A or B or C or D?

- **Multiclass Logistic Regression** ← Fast training times, linear model
- **Multiclass Neural Network** ← Accuracy, long training times
- **Multiclass Decision Forest** ← Accuracy, fast training times
- **One-vs-All Multiclass** ← Depends on the two-class classifier
- **One-vs-One Multiclass** ← Depends on binary classifier, less sensitive to an imbalanced dataset with larger complexity
- **Multiclass Boosted Decision Tree** ← Non-parametric, fast training times and scalable

Predict between two categories

Two-Class Classification

Answers simple two-choice questions, like yes or no, true or false

Answers questions like: Is this A or B?

- **Two-Class Support Vector Machine** ← Under 100 features, linear model
- **Two-Class Averaged Perceptron** ← Fast training, linear model
- **Two-Class Decision Forest** ← Accurate, fast training
- **Two-Class Logistic Regression** ← Fast training, linear model
- **Two-Class Boosted Decision Tree** ← Accurate, fast training, large memory footprint
- **Two-Class Neural Network** ← Accurate, long training times

Generate recommendations

Recommenders

Predicts what someone will be interested in

Answers the question: What will they be interested in?

- **Use the Train Wide & Deep Recommender module** ← Hybrid recommender, both collaborative filtering and content-based approach
- **SVD Recommender** ← Collaborative filtering, better performance with lower cost by reducing dimensionality

Predict values

Regression

Makes forecasts by estimating the relationship between values

Answers questions like: How much or how many?

- **Fast Forest Quantile Regression** ← Predicts a distribution
- **Poisson Regression** ← Predicts event counts
- **Linear Regression** ← Fast training, linear model
- **Bayesian Linear Regression** ← Linear model, small data sets
- **Decision Forest Regression** ← Accurate, fast training times
- **Neural Network Regression** ← Accurate, long training times
- **Boosted Decision Tree Regression** ← Accurate, fast training times, large memory footprint

Discover structure

Clustering

Separates similar data points into intuitive groups

Answers questions like: How is this organized?

- **K-Means** ← Unsupervised learning

Find unusual occurrences

Anomaly Detection

Identifies and predicts rare or unusual data points

Answers the question: Is this weird?

- **One Class SVM** ← Under 100 features, aggressive boundary
- **PCA-Based Anomaly Detection** ← Fast training times

Classify images

Image Classification

Classifies images with popular networks

Answers questions like: What does this image represent?

- **ResNet** ← Modern deep learning neural network
- **DenseNet** ← Modern deep learning neural network

Share this poster: aka.ms/mlcheatsheet

Deployment

Goal

Deploy models with a data pipeline to a production or production-like environment for final user acceptance.

How to do it

The main task addressed in this stage:

- **Operationalize the model**: Deploy the model and pipeline to a production or production-like environment for application consumption.

Artifacts

- A status dashboard that displays the system health and key metrics
- A final modeling report with deployment details
- A final solution architecture document

Customer Acceptance

Goal

Finalize project deliverables: Confirm that the pipeline, the model, and their deployment in a production environment satisfy the customer's objectives.

How to do it

There are two main tasks addressed in this stage:

- **System validation**: Confirm that the deployed model and pipeline meet the customer's needs.
- **Project hand-off**: Hand the project off to the entity that's going to run the system in production.

Artifacts

The main artifact produced in this final stage is the **Exit report of the project for the customer**. This technical report contains all the details of the project that are useful for learning about how to operate the system. TDSP provides an Exit report template. You can use the template as is, or you can customize it for specific client needs.

Different Template to Organize Your Thoughts

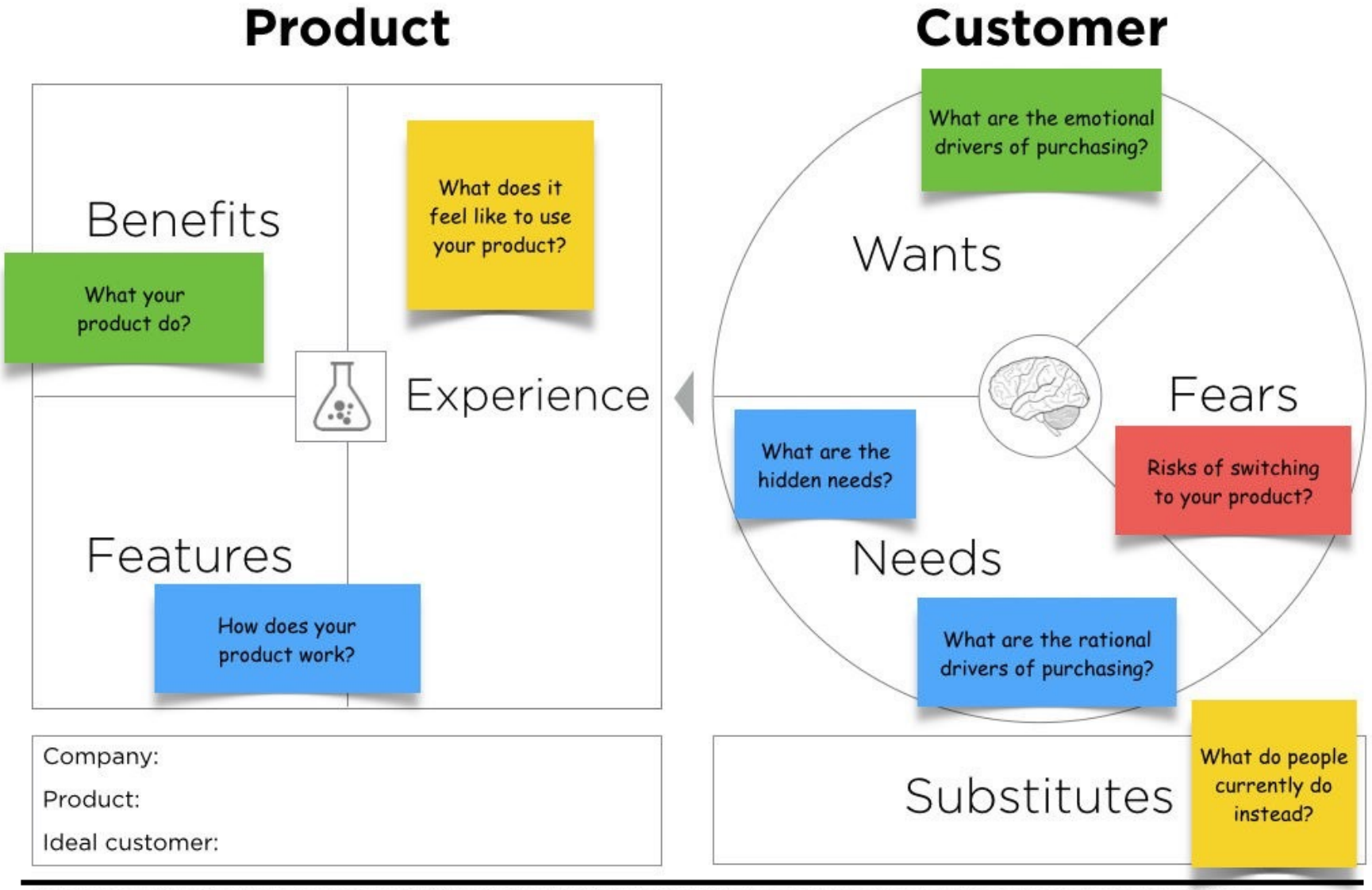
- Value Proposition Canvas
- Business Model Canvas
- AI Canvas
- Machine Learning Canvas

Value Proposition Canvas

A value proposition is the place where your company's product intersects with your customer's desires. It's the magic fit between **what** you make and **why** people buy it. Your value proposition is the crunch point between business strategy and brand strategy.

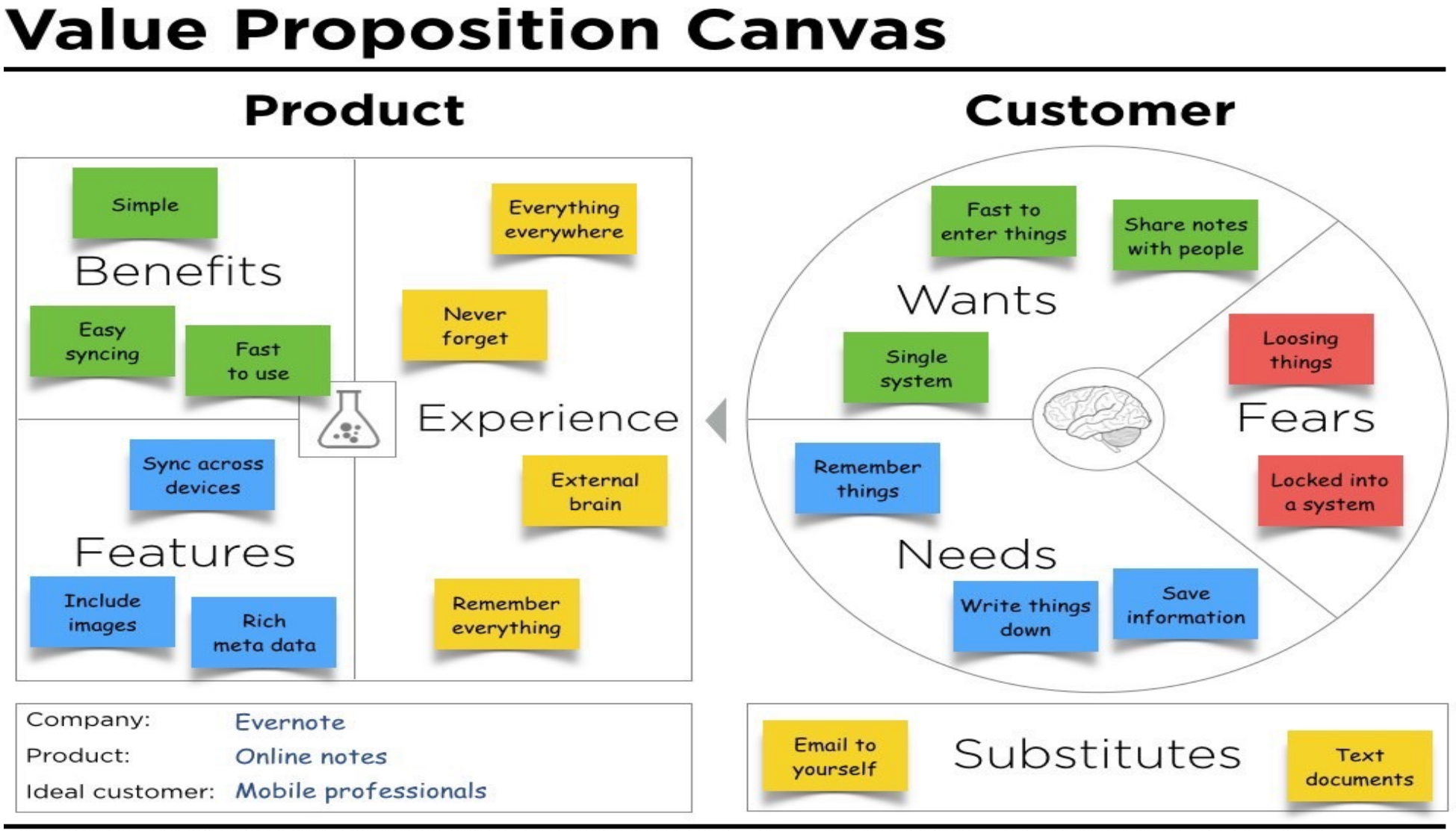
A good match is essential for your business success.

Value Proposition Canvas



Value Proposition Example

Evernote's value proposi(on is translated directly into their marke(ng materials.



Business Model Canvas

- The Business Model Canvas is a chart that maps the key things that a business needs to get right to be successful.
- The Business Model Canvas has become the preferred tool for modern start-ups to use when rapidly testing a new business idea.
- It's an attractive tool because it condenses years of business school and management consulting practise into a single page (with some straight-forward questions for each section).

The Business Model Canvas

Designed for:	Designed by:	Date:	Version:

Key Partners

Who are our Key Partners?
Who are our key suppliers?
Which Key Resources are we acquairing from partners?
Which Key Activities do partners perform?

MOTIVATIONS FOR PARTNERSHIPS
Optimization and economy
Reduction of risk and uncertainty
Acquisition of particular resources and activities

Key Activities

What Key Activities do our Value Propositions require?
Our Distribution Channels?
Customer Relationships?
Revenue streams?

CATERGORIES
Production
Problem Solving
Platform/Network

Key Resources

What Key Resources do our Value Propositions require?
Our Distribution Channels? Customer Relationships?
Revenue Streams?

TYPES OF RESOURCES
Physical
Intellectual (brand patents, copyrights, data)
Human
Financial

Value Propositions

What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?
What bundles of products and services are we offering to each Customer Segment?
Which customer needs are we satisfying?

CHARACTERISTICS
Newness
Performance
Customization
"Getting the Job Done"
Design
Brand/Status
Price
Cost Reduction
Risk Reduction
Accessibility
Convenience/Usability

Customer Relationships

What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

EXAMPLES
Personal assistance
Dedicated Personal Assistance
Self-Service
Automated Services
Communities
Co-creation

Channels

Through which Channels do our Customer Segments want to be reached?
How are we reaching them now?
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are we integrating them with customer routines?

CHANNEL PHASES
1. ***Awareness***
   How do we raise awareness about our company's products and services?
2. ***Evaluation***
   How do we help customers evaluate our organization's Value Proposition?
3. ***Purchase***
   How do we allow customers to purchase specific products and services?
4. ***Delivery***
   How do we deliver a Value Proposition to customers?
5. ***After sales***
   How do we provide post-purchase customer support?

Customer Segments

For whom are we creating value?
Who are our most important customers?

Mass Market
Niche Market
Segmented
Diversified
Multi-sided Platform

Cost Structure

What are the most important costs inherent in our business model?
Which Key Resources are most expensive?
Which Key Activities are most expensive?

IS YOUR BUSINESS MORE
Cost Driven (leanest cost structure, low price value proposition, maximum automation, extensive outsourcing)
Value Driven (focused on value creation, premium value proposition)

SAMPLE CHARACTERISTICS
Fixed Costs (salaries, rents, utilities)
Variable costs
Economies of scale
Economies of scope

Revenue Streams

For what value are our customers really willing to pay?
For what do they currently pay?
How are they currently paying?
How would they prefer to pay?
How much does each Revenue Stream contribute to overall revenues?

TYPES
Asset sale
Usage fee
Subscription Fees
Lending/Renting/Leasing
Licensing
Brokerage fees
Advertising

FIXED PRICING
List Price
Product feature dependent
Customer segment dependent
Volume dependent

DYNAMIC PRICING
Negotiation (bargaining)
Yield Management
Real-time-Market

Strategyzer
strategyzer.com

NETFLIX BUSINESS MODEL

GARYFOX.CO

KEY PARTNERS

- Investors
- Media Producers
- Film Maker Guilds
- Cinemas, Theaters
- TV Networks
- Amazon AWS
- Consumer Electronic Companies
- Regulators

KEY ACTIVITIES

- Technology R&D
- Content licensing
- Content production
- Content distribution
- Data anlaytics
- Sales and marketing

KEY RESOURCES

- Brand
- Apps/website
- Platform
- Employees
- Film Makers/Producers
- Prizes/Awards

VALUE PROPOSITIONS

- 24/7 On Demand Entertainment
- View high-definition shows and movies
- Stream content
- Unlimited access
- Netflix Orginal
- 30 Day free trial
- No commercials

CUSTOMER RELATIONSHIPS

- Self service
- On-demand
- Ease of use

CHANNELS

- Any Device
- Netflix App
- Word of mouth
- Online advertising
- Offline advertising
- Social Media

CUSTOMER SEGMENTS

Micro-segmentation
- 2000 preference clusters

Usage
- usage segmentation

Geographical
- content/languages

COST STRUCTURE

- Production
- Research and Development
- Licensing
- Infrastructure - AWS
- Marketing
- Payment Processing Fees
- General/Admin

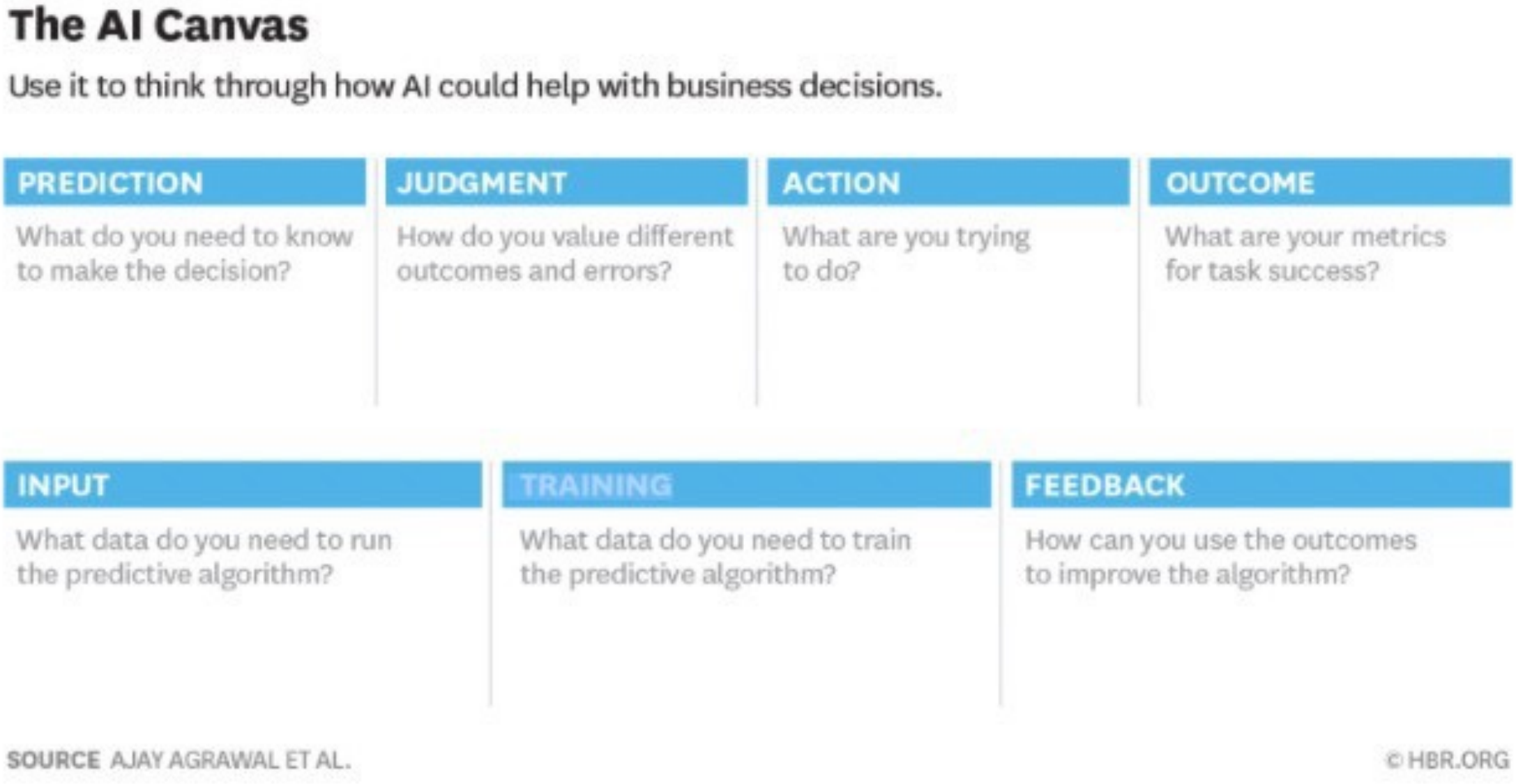
REVENUE STREAMS

- Subscription Model
- Product Placement
- DVD Rental
- Future Model - licensing Netflix owned content

Netflix Business Model: www.garyfox.co/netflix-business-model/

AI Canvas

how do you think through what it would take to incorporate a prediction machine into your decision-making process?



AI Canvas Example

Example: Over 97% of the time that a home security alarm goes off, it's a false alarm. That is, something other than an unknown intruder (threat) triggered it. This requires security companies to make a decision as to what to do: Dispatch police or a guard? Phone the homeowner? Ignore it?

If the security company decides to take action, more than 90 out of 100 times, it will turn out that the action was wasted. However, always taking an action in response to an alarm signal means that when a threat is indeed present, the security company responds.
How can you decide whether employing a prediction machine will improve matters? The AI Canvas is a simple tool that helps you organize what you need to know into seven categories in order to systematically make that assessment.

The AI Canvas: An Example Using AI to Improve Home Security

PREDICTION	JUDGMENT	ACTION	OUTCOME
Predict whether an alarm is caused by an unknown person vs. something else (i.e., true vs. false).	Compare the cost of responding to a false alarm to the cost of not responding to a true alarm.	Dispatch a security response or not when an alarm is triggered.	Observe whether the action taken in response to the triggered alarm was correct.

INPUT	TRAINING	FEEDBACK
Sensor inputs from movement, heat, camera, and contextual data at each point in time when the alarm is on; these data are used to operate the AI.	Historical sensor data matched with historical outcome data (actual intruder vs. false alarm); these data are used to train the AI before it is deployed.	Sensor data matched with data collected from outcomes (verified intruders vs. verified false alarms); these data are used to update the model, continuously improving the AI while it is operating.

SOURCE AJAY AGRAWAL ET AL.

© HBR.ORG

AI Project Canvas Title:

Data
Which data do you need?

Skills
Which skills do you need for development?

Output
Which key metric are you optimizing for?

Value Proposition
What is the value added by your project?

Integration
How will the project be integrated?

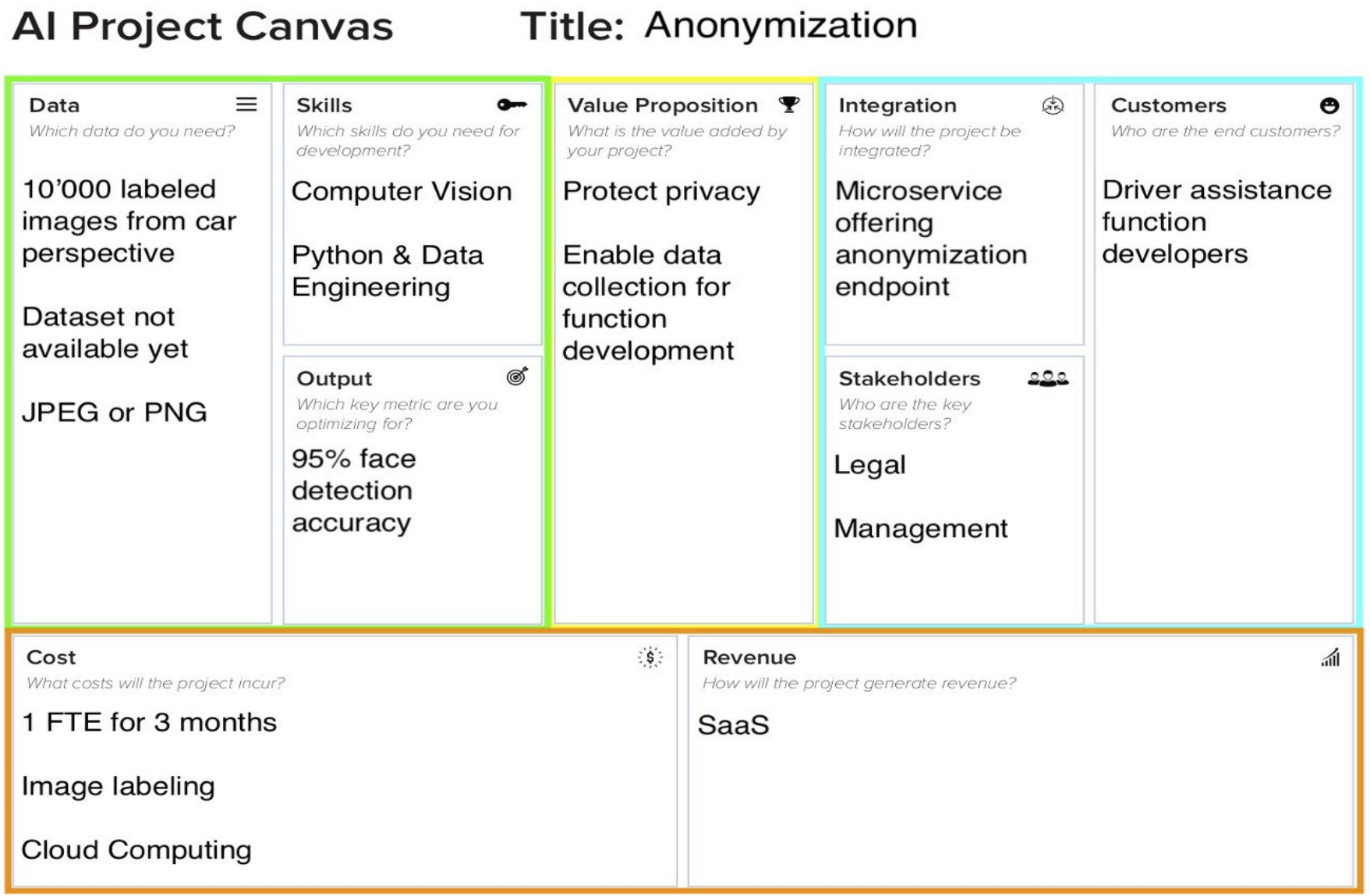
Stakeholders
Who are the key stakeholders?

Customers
Who are the end customers?

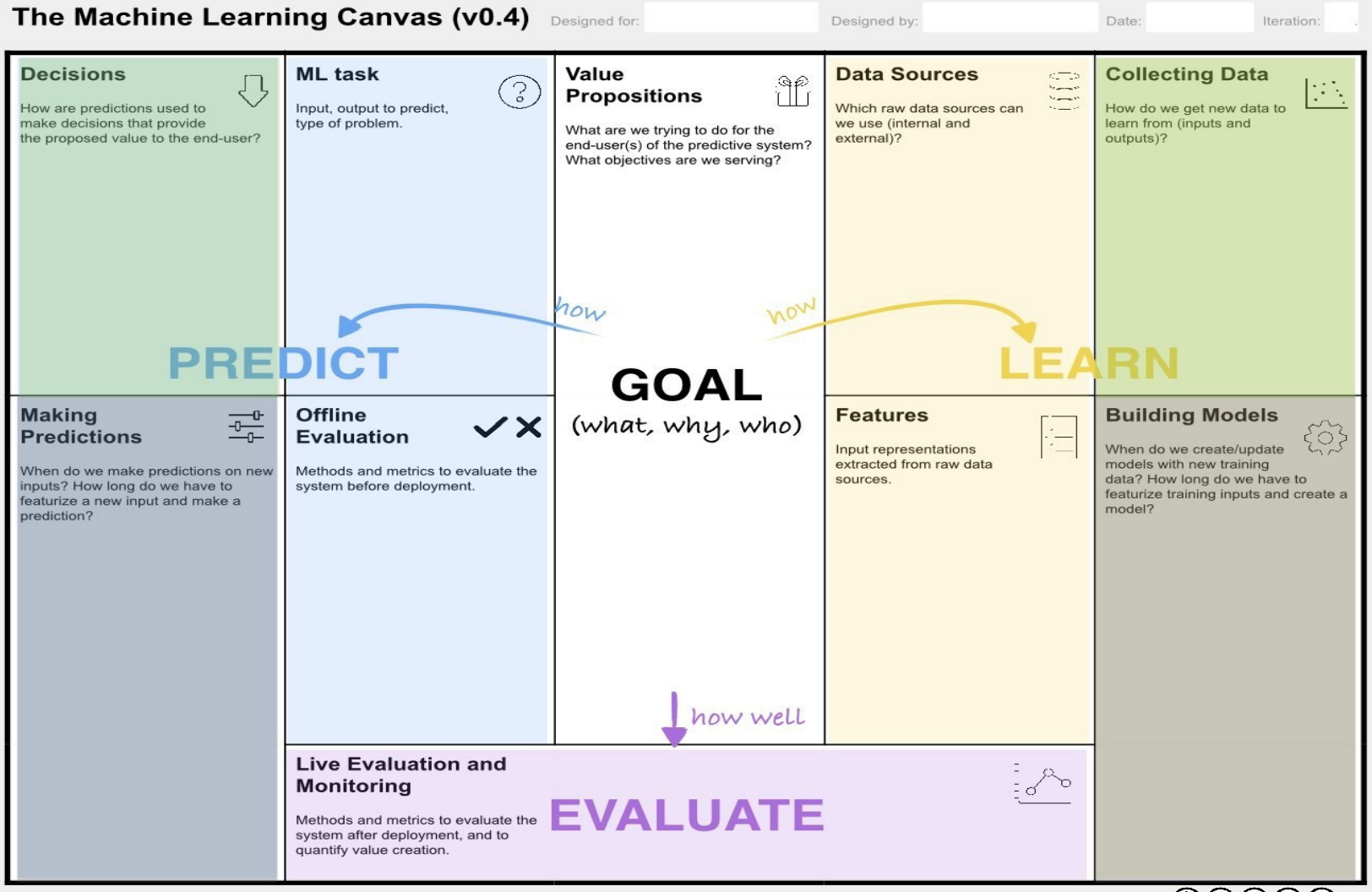
Cost
What costs will the project incur?

Revenue
How will the project generate revenue?

AI Project (AI + Business Model) Canvas



Machine Learning Canvas



background

specifics

Domain Integration

Predictive Engine

Example: Churn perdition for CRM customers

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?

On 1st day of every month:

- Randomly filter out 50% of customers (hold-out set)
- Filter out 'no-churn'
- Sort remaining by descending (churn prob.) x (monthly revenue) and show prediction path for each
- Target as many customers as suggested by simulation

ML task

Input, output to predict, type of problem.

Predict answer to "Is this customer going to churn in the coming month?"

Input : Customer

Output: "churn" or "no-churn" ("churn" is the positive class)

⇒ Binary Classification

Value Propositions

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

- **Context:**
- Company sells SaaS with monthly subscription
- End-user of predictive system is CRM team
- **We want to help them…**
- Identify important clients who may churn, so appropriate action can be taken
- Reduce churn rate among high-revenue customers
- Improve success rate of retention efforts by understanding why customers may churn

Data Sources

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

- CRM tool
- Payments database
- Website analytics
- Customer support
- Emailing to customers

Collecting Data

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

-Every month, we see which of last month's customers churned or not, by looking through the payments database.

-Associated inputs are customer "snapshots" taken last month.

Making Predictions

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

Every month we (re-)featurize all current customers and make predictions for them.

We do this overnight (along with building the model that powers these predictions and evaluating it).

Offline Evaluation

Methods and metrics to evaluate the system before deployment.

Before targeting customers:

- Evaluate new model's accuracy on pre-defined customer profiles
- Simulate decisions taken on last month's customers (using a model learnt from customers 2 months ago). Compute ROI w. different # customers to target & hypotheses on retention success rate (is it >0?)

Features

Input representations extracted from raw data sources.

Basic customer info at time t: age, city, etc.

Events between (t - 1 month) and (t):

- Usage of product: # times logged in, functionalities used, etc.
- Customer support interactions
- Other contextual, e.g. devices used

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?

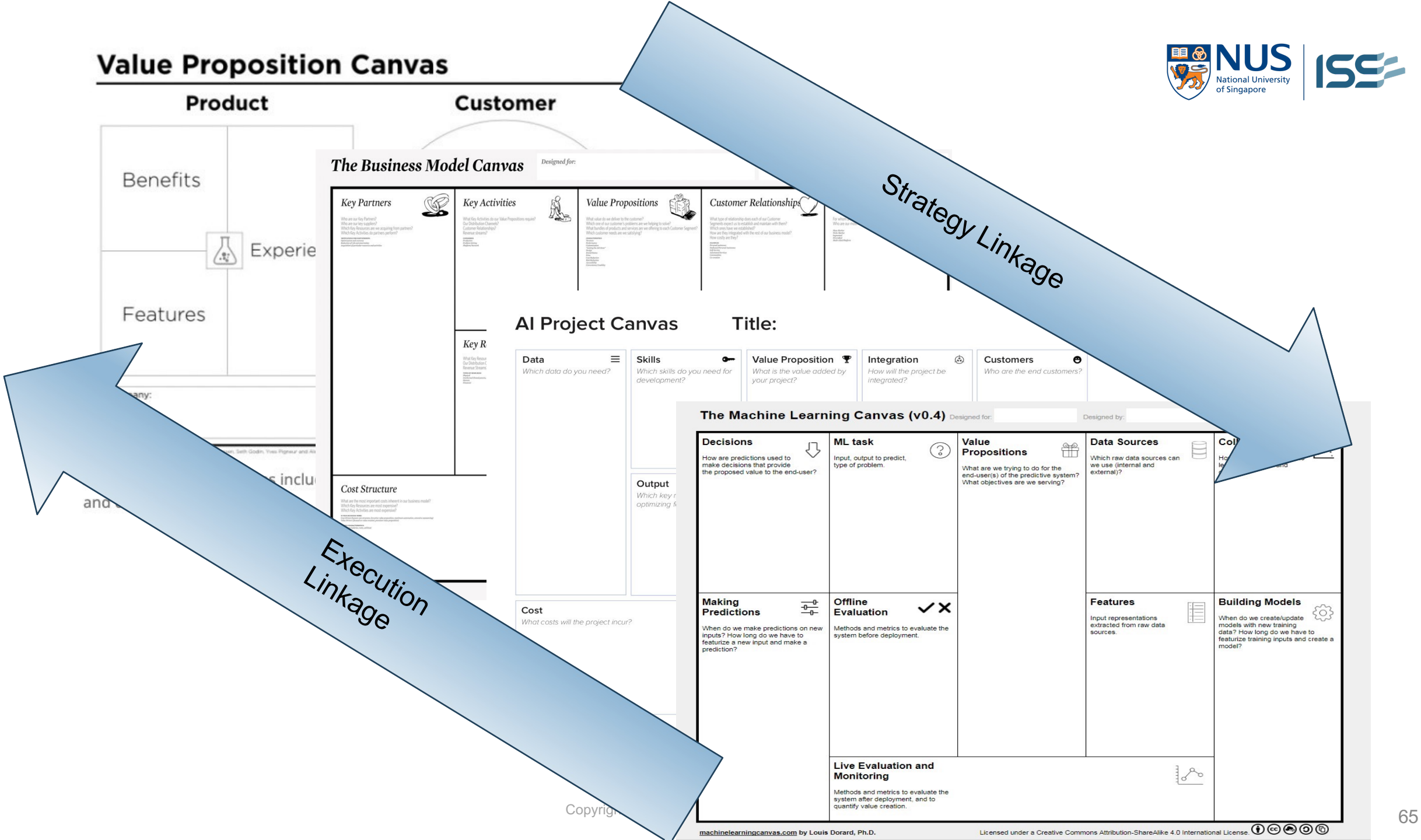
Every month we create a new model from the previous month's hold-out set (or the whole set, when initializing this system).

We do this overnight (along with offline evaluation and making predictions).

Live Evaluation and Monitoring

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

- Accuracy of last month's predictions on hold-out set
- Compare churn rate & lost revenue between last month's hold-out set and remaining set
- Monitor (#non-churn among targeted) / #targets
- Monitor ROI (based on difference in lost revenue & cost of retention campaign)


Value Proposition Canvas
Product
Customer
Benefits
Features
The Business Model Canvas
Designed for:
Key Partners
Key Activities
Value Propositions
Customer Relationships
Cost Structure
AI Project Canvas
Title:
Data
Which data do you need?
Skills
Which skills do you need for development?
Value Proposition
What is the value added by your project?
Integration
How will the project be integrated?
Customers
Who are the end customers?
Output
Cost
What costs will the project incur?
The Machine Learning Canvas (v0.4)
Designed for:
Designed by:
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)?
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.
Licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
Strategy Linkage
Execution Linkage

Estimate the ROI for implementing an AItiML tool to perform each task

ROI = Gain from Investment / Cost of Investment

Gain from Investment

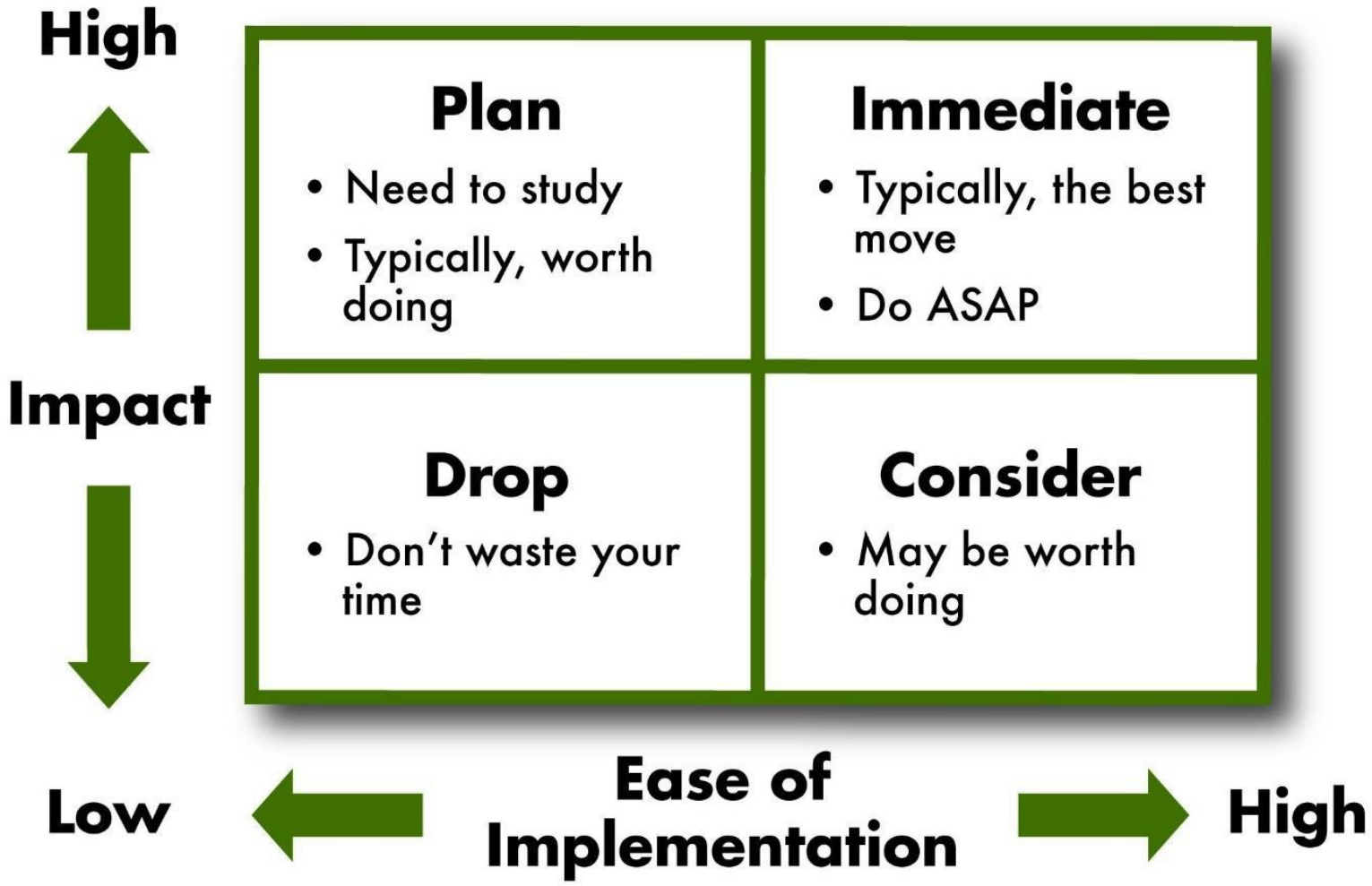
- Revenue Optimization
  I. Existing Revenue Stream:
  II. New Revenue Stream:
- Cost Optimization
  - Human Capital Gain = (Employee Average Cost per Hour x Total Number of Increased Productivity hours)

Cost of Investment

- CAPEX: Non-recurring cost
- OPEX: Ongoing cost

ROI = (Revenue Optimization + Cost Optimization) / (CAPEX + OPEX)

Ease of Implementation Vs Impact



Make or Buy Analysis

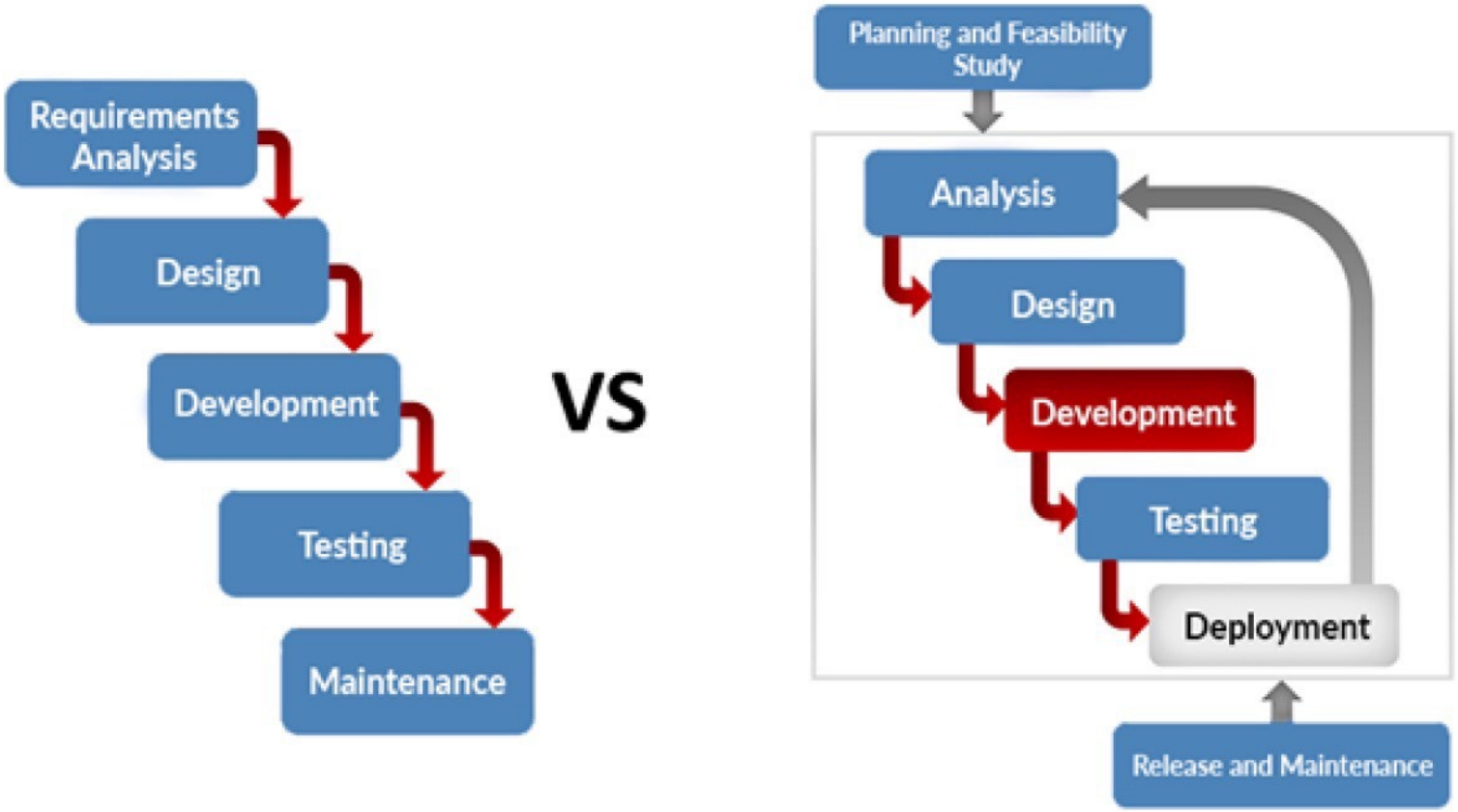
Option 1: Build In-House	Option 2: Buy "Out of the Box" Software	Option 3: Hybrid Build & Buy (Recommended)	Option 4: Do Nothing
Benefits Saves money. Gives us control over build.	**Benefits** Benefit from industry standards. Less time needed.	**Benefits** Saves some time and money. Gives us control.	**Benefits** Saves time and money.
Costs Time. Pushes back other priorities.	**Costs** Upfront cost of purchase. Support costs.	**Costs** Some time and money. Bad user experience.	**Costs** Operational costs.
Risks No guarantee we will do a better job than OTB.	**Risks** May need more customisation at high cost.	**Risks** More gaps in bought product than expected.	**Risks** Unable to scale: could be difficult to adapt to market.

Workflow Design

Agile
Scrum
Kanban
Data Driven Scrum

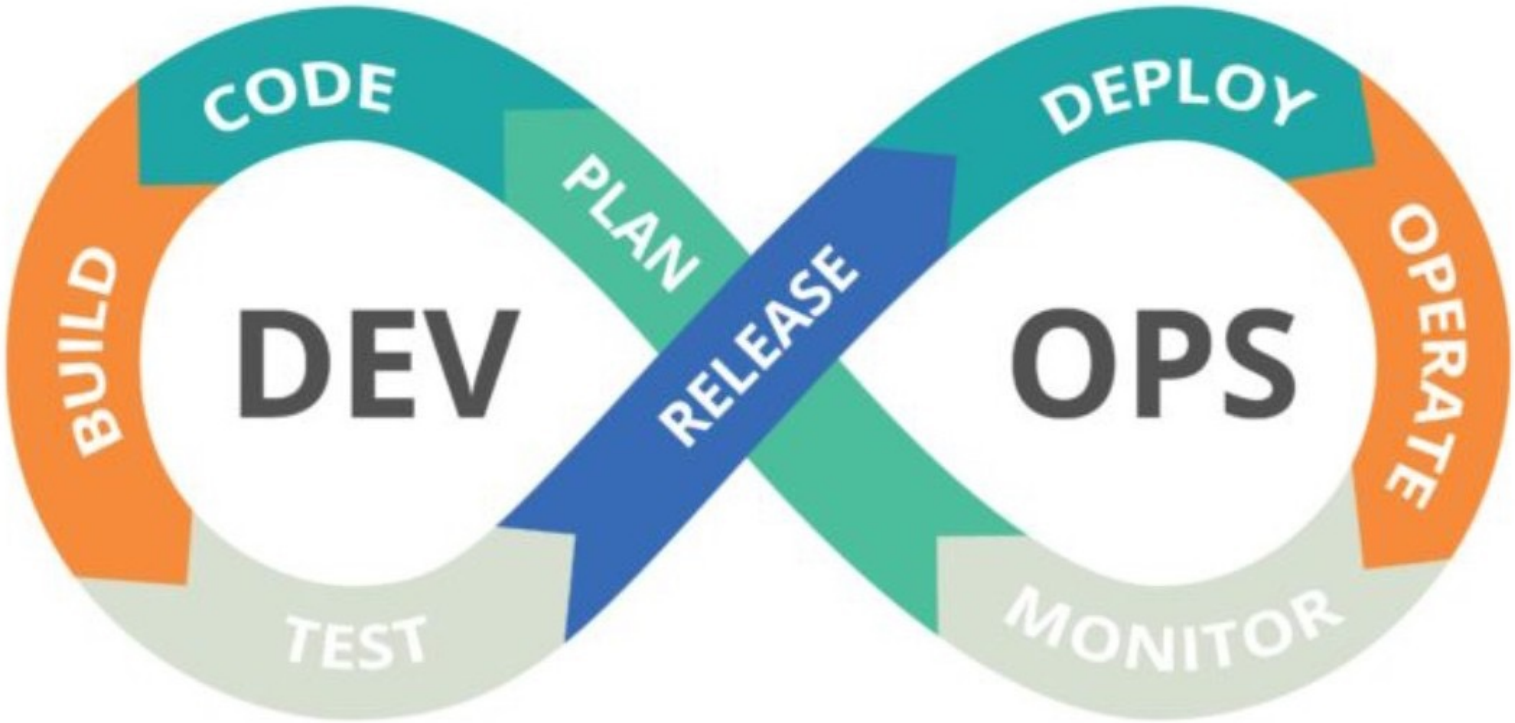
Concepts and Contexts

Waterfall & Agile



DevOps

DevOps: Faster version of Agile through continuous integration, continuous deployment and continuous delivery.

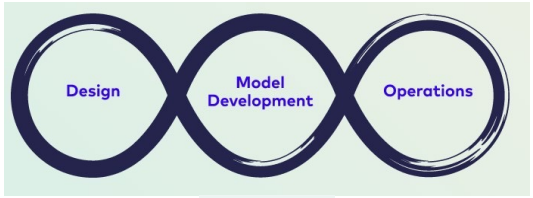


In Software Development Context

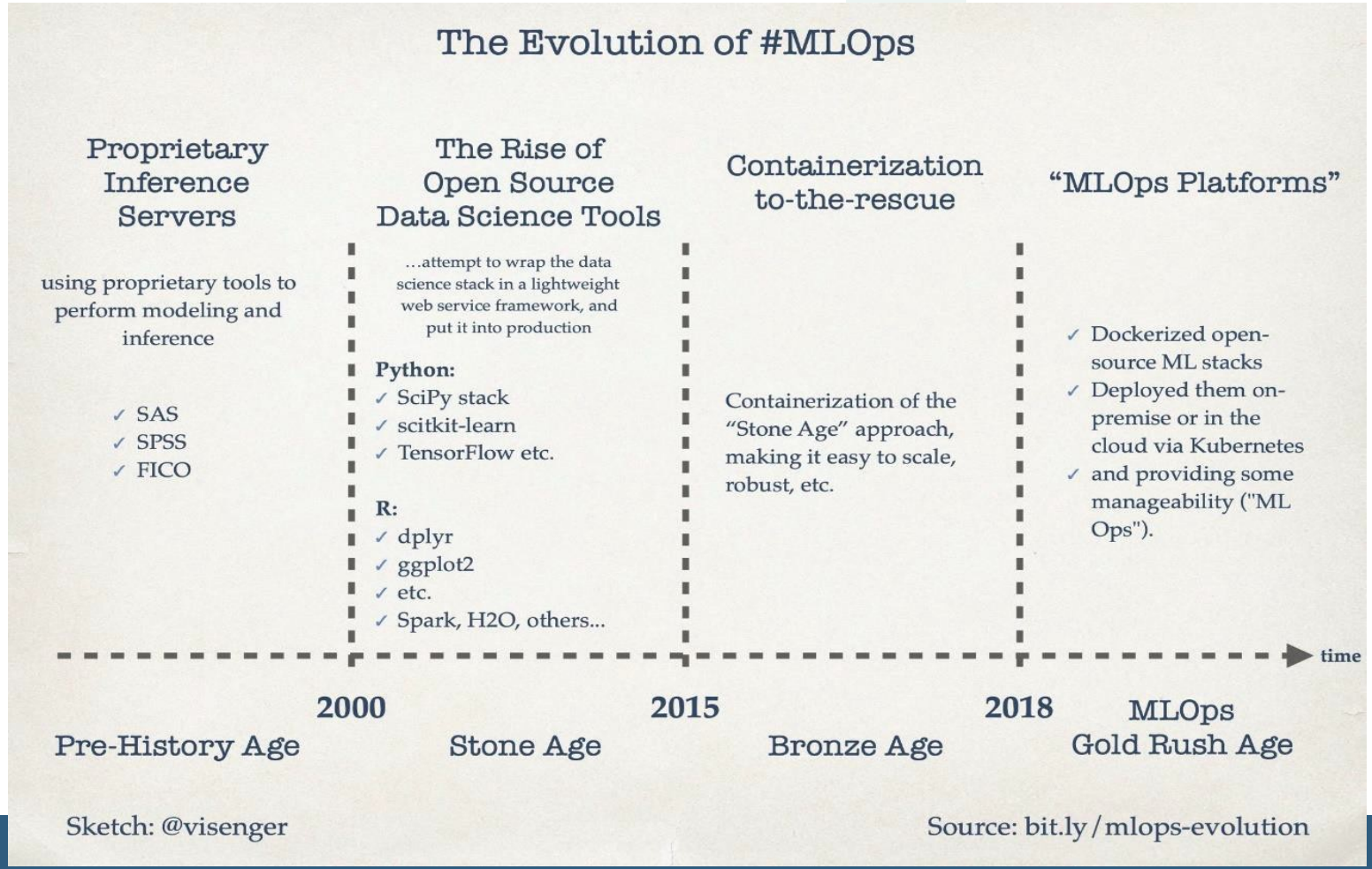
Agile	DevOps
Feedback from Customer	Feedback from Self
Small & Fast Release Cycle	Small & Fast Release Cycle + Immediate Feedback
Speed is the Key	Speed & Automation are the Keys

https:titimedium.comticlaruswaytipopular-devops-tools-review-ee0cffea14ec

The Evolution of ML Operations



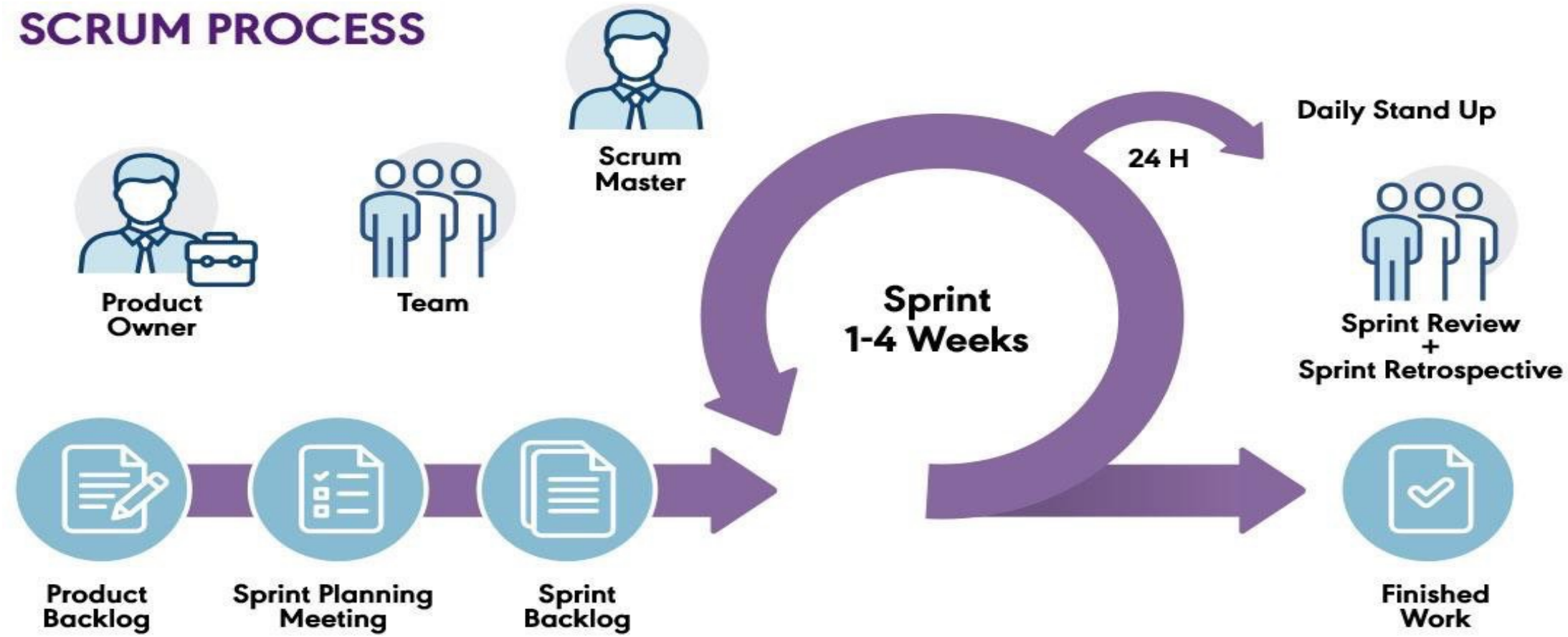
MLOps



The term MLOps is defined as "the extension of the DevOps methodology to include Machine Learning and Data Science assets as first-class citizens within the DevOps ecology" ***Source: MLOps SIG.***

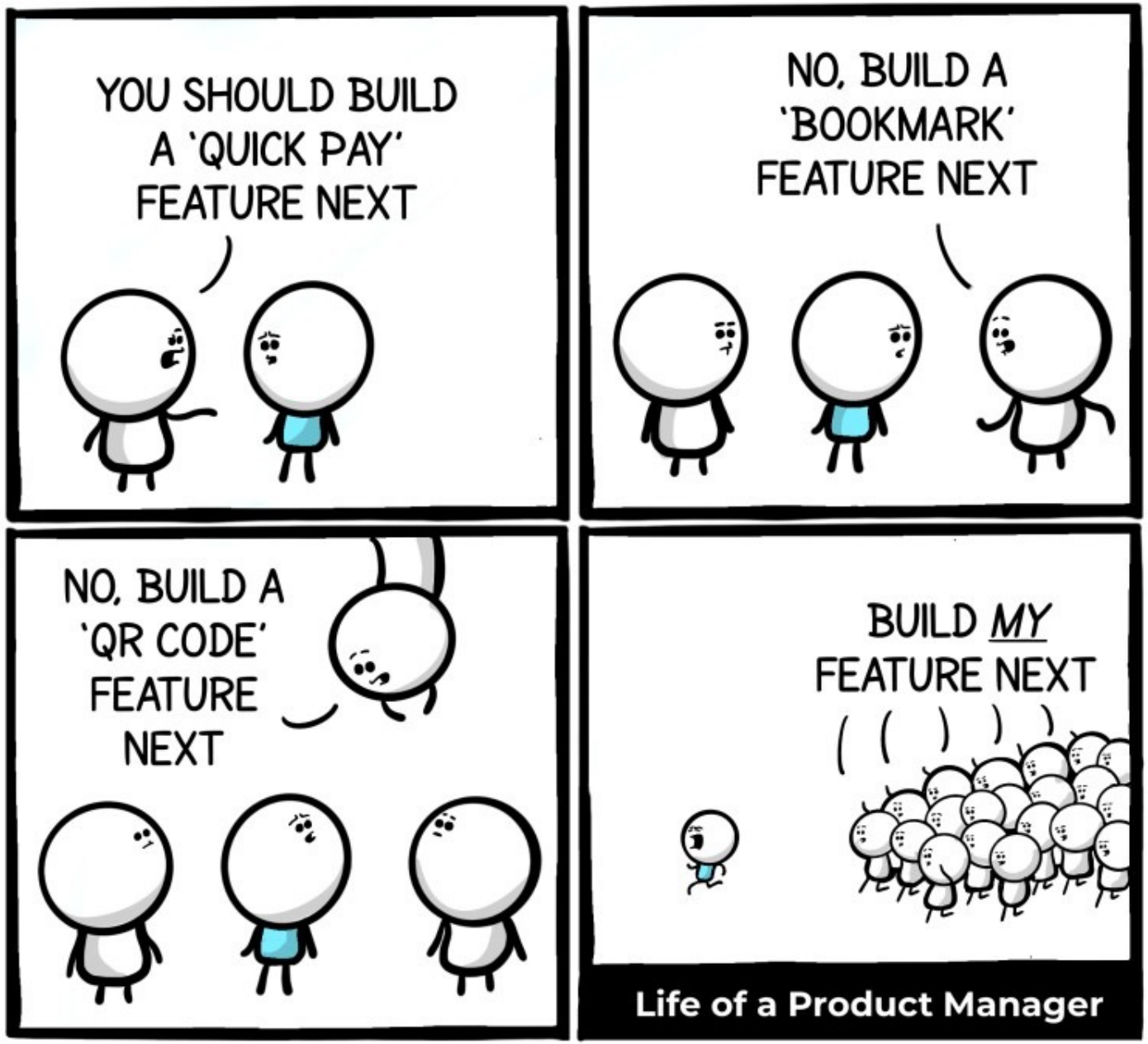
MLOps, like DevOps, emerges from the understanding that separating the ML model development from the process that delivers it — ML operations — lowers quality, transparency, and agility of the whole intelligent software.

Scrum



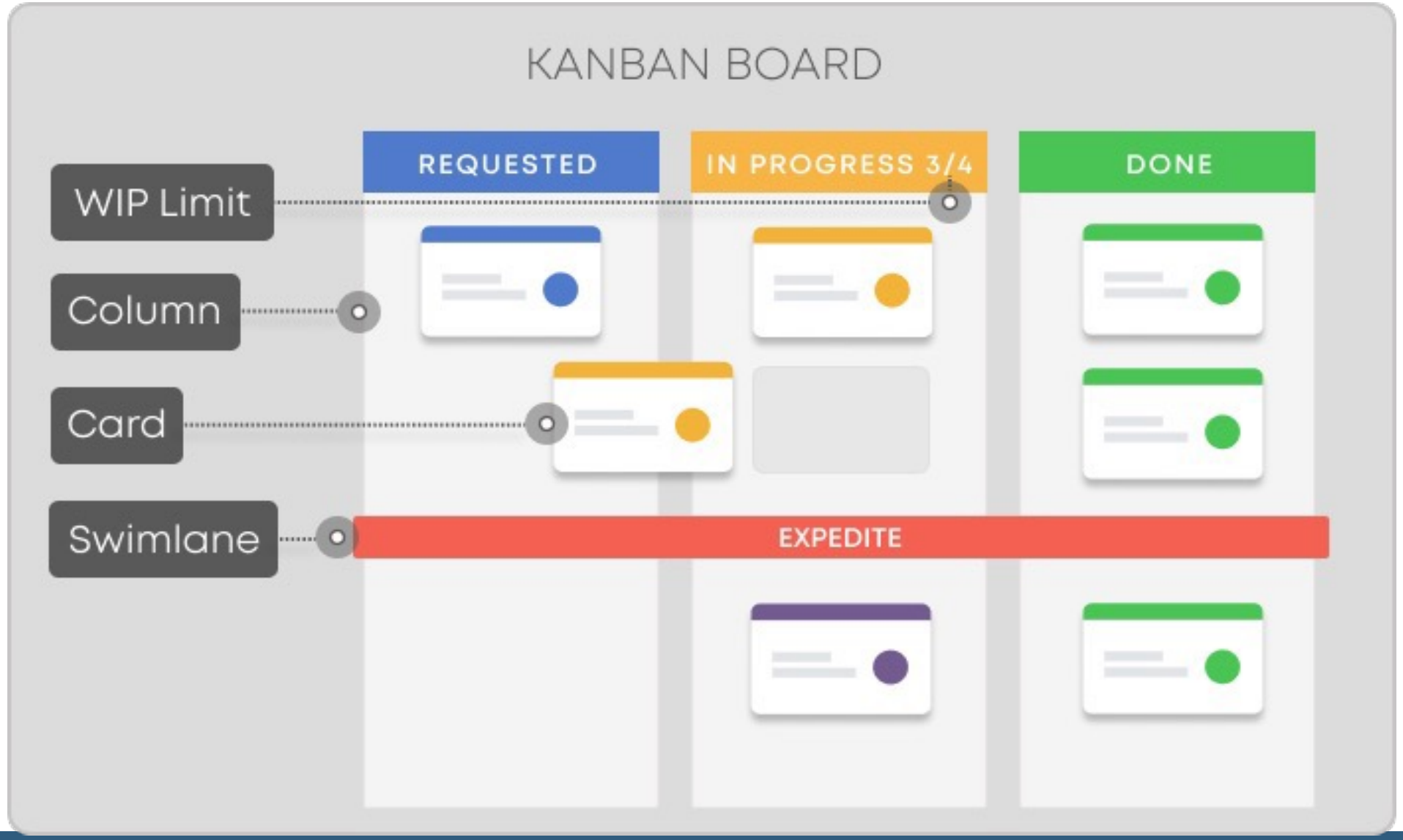
Responsibilities of Scrum Master

- Clear Obstacles
- Establish a productive environment
- Address team dynamics issues if there is any
- Ensure a good relationship between the team and product owner and outside stakeholders
- Protect the team from outside interruptions and distractions



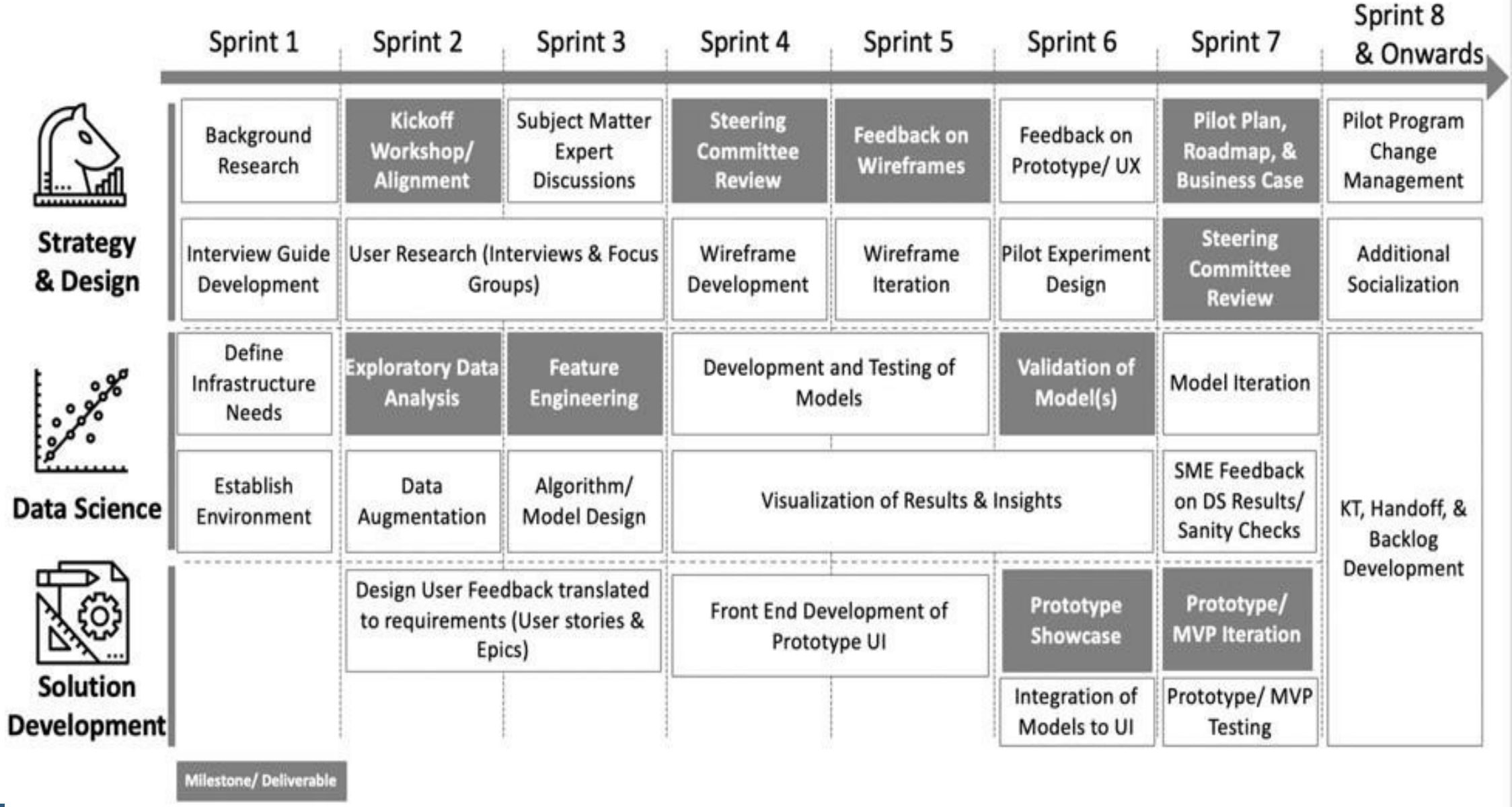
Kanban– A pull system

Two Principles of Kanban: Visualize your work and Limit Work-in-Progress



	Scrum	Kanban
Origin	Software development	Lean manufacturing
Idealogy	Learn through experiences, self-organize and prioritize, and reflect on wins and losses to continuously improve.	Use visuals to improve work-in-progress
Cadence	Regular, fixed-length sprints (i.e. two weeks)	Continuous flow
Practices	Sprint planning, sprint, daily scrum, sprint review, sprint retrospective	Visualize the flow of work, limit work-in-progress, manage flow, incorporate feedback loops
Roles	Product owner, scrum master, development team	No required roles

What this looks like in practice



Data Pipeline

Data Pipelines

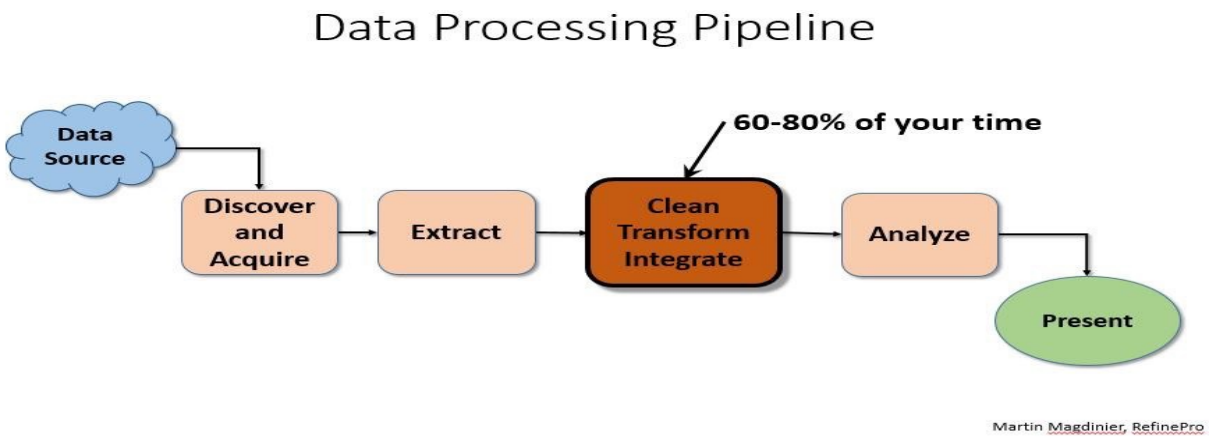
- Data pipelines are
  - sets of processes that move and transform data from various sources to a destination where new value can be derived.
- They are the foundation of analytics, reporting, and machine learning capabilities.
- Complexity of a data pipeline depends on the following:
  - size, state, and structure of the source data
  - business users' requirements
  - the needs of the analytics project

Who builds data pipelines?

- Data engineers specialize in building and maintaining the data pipelines
- Data engineers work closely with data scientists and analysts to understand what will be done with the data and help bring their needs into a scalable production state.
- Some common skills that all good data engineers possess:
  - SQL and Data Warehousing Fundamentals
  - Python and/or Java
  - Distributed Computing
  - Basic System Administration

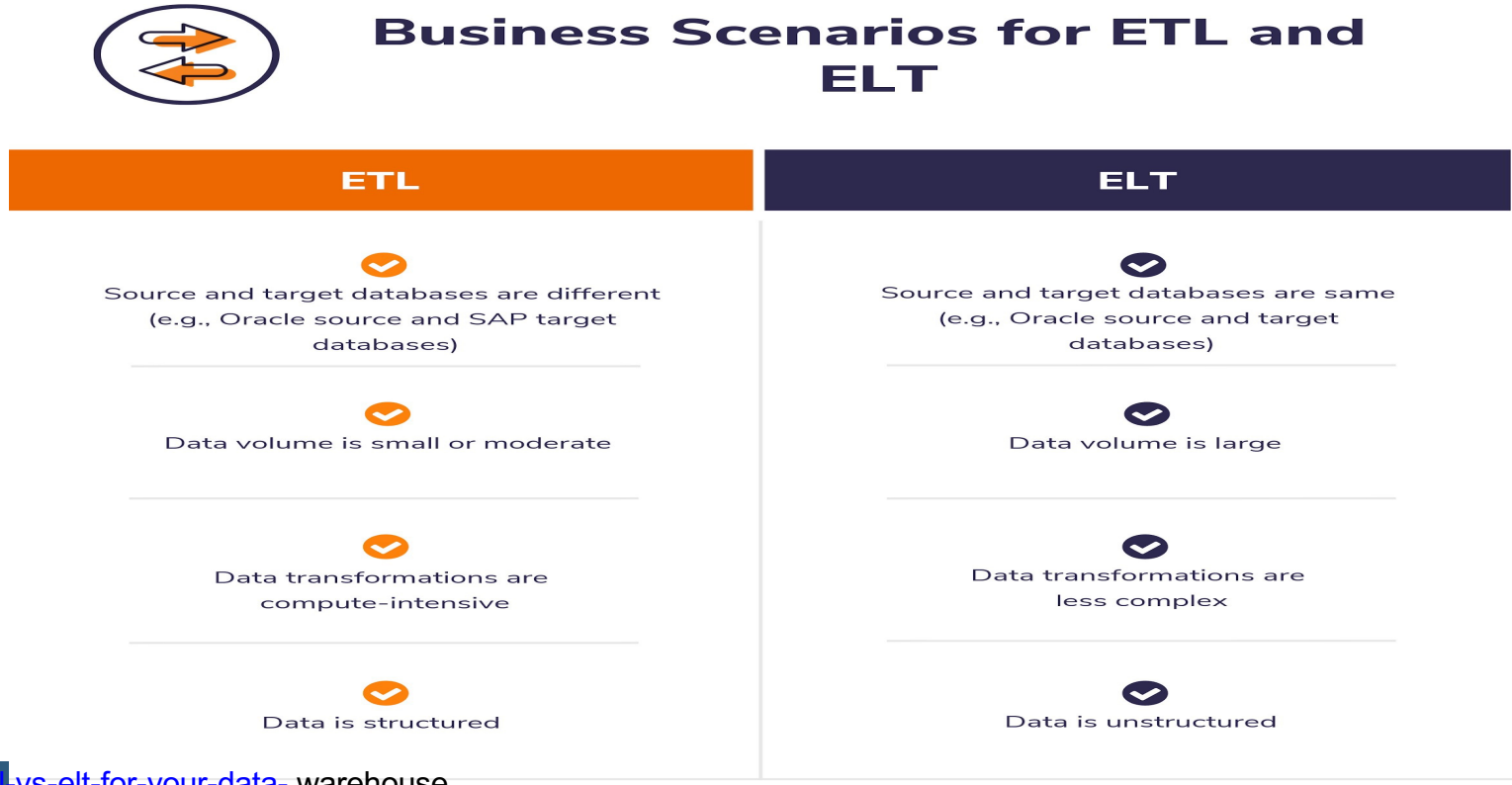
Importance of Data pipelines

- Data pipelines:
  - Work behind the scene for every dashboard and insight that a data analyst generates and for each predictive model developed by a data scientist
  - Refine raw data along the way to clean, structure, normalize, combine, aggregate, and at times anonymize or otherwise secure it.
  - Keep data flowing to solve problems, inform decisions.



Common Data Pipeline Patterns

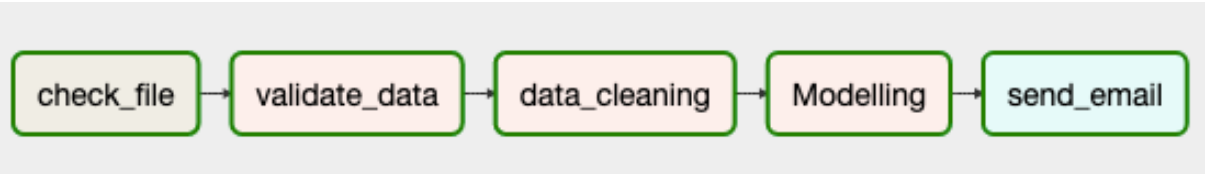
- **ETL and ELT**:
  - Both patterns are approaches to data processing used to feed data into a data warehouse and make it useful to analysts and reporting tools.
  - The difference between the two is the order of their final two steps (transform and load),
  - **ELT for Data Analysis**
  - **ELT for Data Science**
  - **ELT for Data Products and Machine Learning**

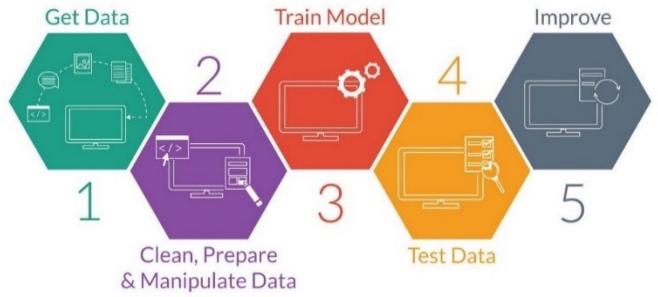


https://www.softwareadvice.comtiresourcestietl-vs-elt-for-your-data- warehouse

Common Data Pipeline Patterns

- Steps in Machine Learning Pipeline
  - Data ingestion
  - Data validation
  - Data preprocessing
  - Model training
  - Model deployment
- Incorporate Feedback in the Pipeline
  - Any good ML pipeline will also include gathering feedback for improving the model



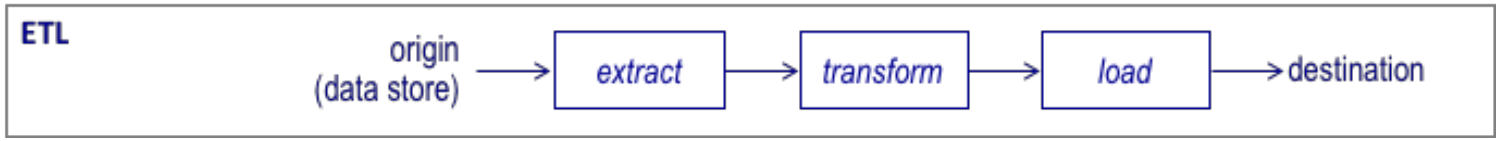


Key characteristics of a data pipeline

- Data frequency:
  - small batches or real time.
  - maintain the frequency of data transfer
- Resiliency:
  - Should be fault tolerant and resilient
  - no loss of data.
- Scalability:
  - re-configurable to scale out onto more hardware nodes if the data load increases.

Data Pipeline Architectures

- **Batch Data Pipeline:**
  - Batch data pipelines move large sets of data at a particular time or in response to a behavior or when a threshold is met.
  - A batch data pipeline is often used for bulk ingestion or ETL processing.
  - A batch data pipeline might be used to deliver data weekly or daily from a CRM system to a data warehouse for use in a dashboard for reporting and business intelligence.



Ref : https://iwww.eckerson.comtiarticlestidata-pipeline-design-patterns

Data Pipeline Architectures

- **Streaming Data Pipeline**
  - flow data continuously from source to destination as it is created.
  - used to populate data lakes or as part of data warehouse integration, or to publish to a messaging system or data stream.
  - also used in event processing for real-time applications. For example, streaming data pipelines might be used to provide real-time data to a fraud detection system and to monitor quality of service.

Pipeline challenges

- Speed can be a challenge for both the ingestion process and the data pipeline
- Knowing whether an organization truly needs real-time processing is crucial for making appropriate architectural decisions about data ingestion.
- Legal and compliance requirements add complexity (and expense) to the construction of data pipelines

Tools & Workshops

ML Flow Workshop (focus on tracking)

- An open source platform for the machine learning lifecycle

Swifter

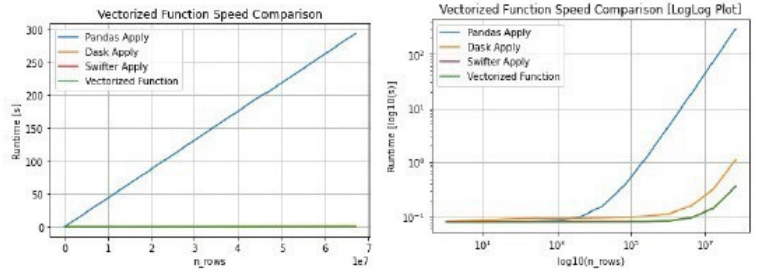
- Open your Google Colab notebook and follow the instructions

swifter

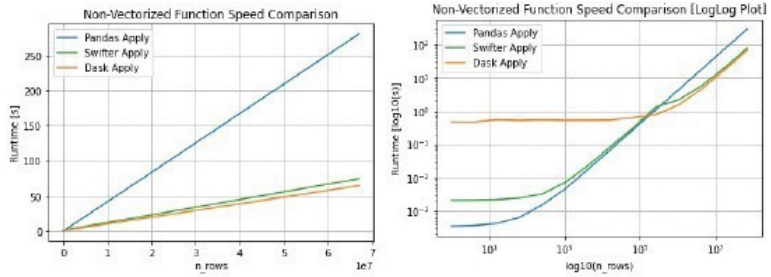
A package which efficiently applies any function to a pandas dataframe or series in the fastest available manner.

pypi package 1.0.9 | circleci passing | codecov 94% | code style black | stars 1.7k | downloads 88k/month

Vectorizes your function, when possible



When vectorization is not possible, automatically decides which is faster: to use dask parallel processing or a simple pandas apply



Great Expectations Workshop

- **Always know what to expect from your data– A data quality monitoring tool**

Great Expectations is a shared, open standard for data quality. It helps data teams eliminate pipeline debt, through data testing, documentation, and profiling.

great_expectations

Case Study--Background

- **HEINEKEN** is one of the leading brewing companies in the world. Led by the Heineken® brand, the group has a portfolio of more than 300 international, regional, local and specialty beers and ciders. It employs over 85,000 employees and operates breweries, malteries, cider plants and other production facilities in more than 70 countries.
- **Global Analytics** is part of the HEINEKEN Digital & Technology function who support HEINEKEN's operating companies by developing tools, creating machine learning models, setting guidelines, and functioning as a "Centre of Excellence" for data analytics.
- The team's focus is to build scalable products to be implemented many times. This is preceded by proving the value of the use cases they develop through experimentation and feasibility studies.

Case Study--The Challenge

- During the experimentation phase, the Global Analytics team deals with a **variety of data from different internal and external sources**, including the company's Enterprise Resource Planning system, sales data, marketing and media data, and even weather data.

- With HEINEKEN being a federated company, the team integrates data coming from the regional operating companies in a multitude of ways, such as SQL access to an on-prem database, flat files that are uploaded on a regular basis, or via an API for weather data.

- Once a new approach to analyzing the data has proven valuable through experimentation, they then develop production data pipelines to scale these efforts globally.

Case Study—Solution

- The Global Analytics team decided to use Great Expectations to validate their incoming data to standardize how validation was done across different engineers and data sets - **no more re-implementation of the same tests** by different people!

- Even though they considered the learning curve of Great Expectations to be fairly steep, they decided to invest in deploying it in their pipelines based on the functionality it provides, as well as the fact that it is an open-source library.

Case Study – Solution

- **The group working with the company's commerce and marketing data implemented Great Expectations in a particularly novel way**: They integrated Great Expectations into a data uploading tool, which is used by upstream data providers to upload Excel files that are fed into machine learning models.

- The tool creates a Pandas dataframe of the uploaded data and runs validation using the Great Expectations. The tool then **instantly provides feedback** to the upstream data provider based on the validation result and accepts or rejects the uploaded file.

- This is a great example of how Great Expectations can be easily integrated into other tools to provide data validation, outside of the typical data pipeline use cases we usually see.

What is Apache Airflow

- Apache Airflow is an open source platform to programmatically author, schedule and monitor workflows.
- It is the one of the best orchestrator to run your data pipeline in the right way following the right order at the right time.
- Airflow is
  - Dynamic (e.g. anything you can do in Python, you can do in airflow with the python executor)
  - Scalable
  - User friendly (e.g their UI is quite intuitive)
  - Eco system friendly (e.g. you can create your own plug ins to interact with the new tools) –check out the link for tools interacting with airflow

What Airflow is not

- Airflow is not designed to execute any workflows directly inside of airflow, but just to schedule them and to keep the execution within the external systems.
  - E.g.
    - submit a spark job and store data on a Hadoop cluster
    - Execute some sql transformation in snowflake
    - Trigger a Sage Maker training job

- Airflow is not a data streaming solution

Airflow Demo (optional)

- Create a folder called “airflow” on your desktop.
- Open that folder via VS code.
- Go to “New Terminal”
- Type “docker --version” and ”docker-compose --version” to confirm docker is there.
- Run the following command
  - curl -LfO 'https://airflow.apache.org/docs/apache-airflow/2.3.4/docker-compose.yaml’
  - **Note: if you are using windows machine, you need to run "remove-item alias:curl" first**
- Run “docker-compose up airflow-init”
- Run “docker-compose up”
- Go to “localhost:8080” and type “airflow” for both username and password
- After finishing, open another terminal and type “docker-compose down” to shut down everything

Great_Expectation Lab

- use avocado data set provide and check the following:
  - whether "Date" column exists
  - Whether there is no null value in the 'region' column
  - whether the "region" column type is string
  - whether date format follows "%Y-%m-%d"
  - whether the average price is between 0.5 and 3
  - whether the 'type' column has two distinct values, namely 'conventional', 'organic'
- Make sure you include all the checkings, including those failed ones into your json file.
- Upload your json file into Canvas (First_Name_Last_Name_Day2.json)


Thank You