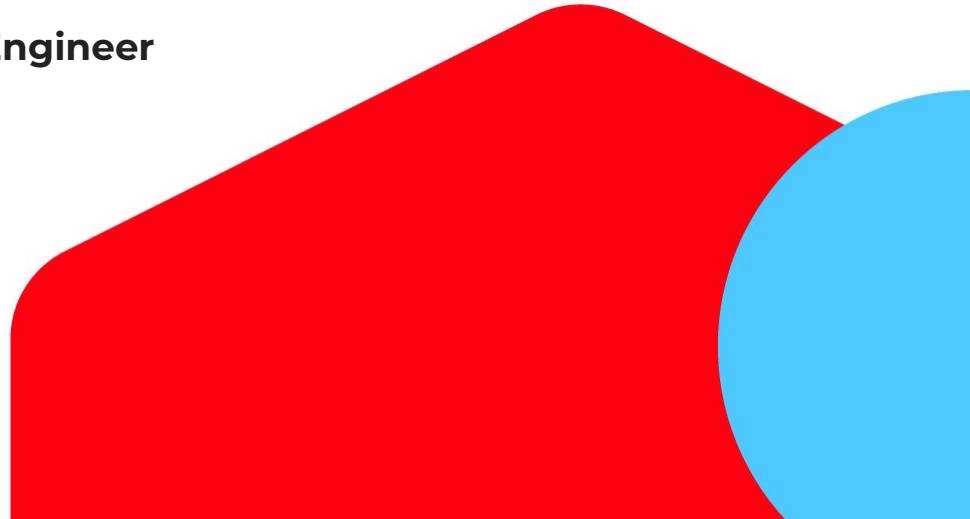
Introduction to Machine Learning

Yusuke Shido
Mercari Recommendation Team / Software Engineer



mercari
DevDojo

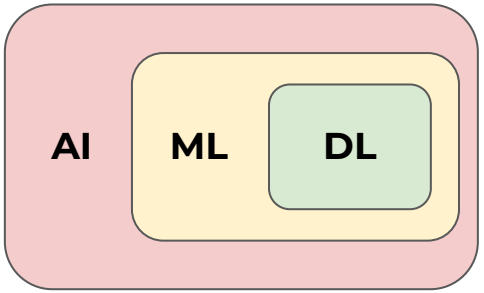
Lecture Overview

- Key Ideas in ML
  - AI and ML
  - ML Basics
  - Preprocessing
- ML at Mercari JP
  - Data at Mercari
  - ML projects applied in different domains

Key Ideas in Machine Learning

AI and ML

- **AI: Artificial Intelligence**
  - Software or computer programs that reproduce human's intellectual activities
  - ex. Recommending items that has specific word in the title
- **ML: Machine learning**
  - **One of the methods** to implement AI
  - We often call non-ML methods as "rule-based method" or "statistical method"
  - ex. Recommending items using an ML model trained using user context and purchases
- **Deep Learning**
  - **One of the methods** to implement ML
  - ML using **deep** neural networks
  - Recently people use "AI" to refer to advanced DNN
  - ex. Recommending items using a neural network



Ref)
https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf

Machine Learning Basics

- Most ML models are trained like:

$$\text{minimize} \quad \text{loss}(f(x;\theta), y) + g(\theta)$$

mercari

Ref)
https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf

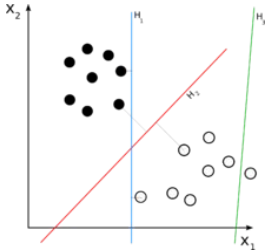
Machine Learning Basics

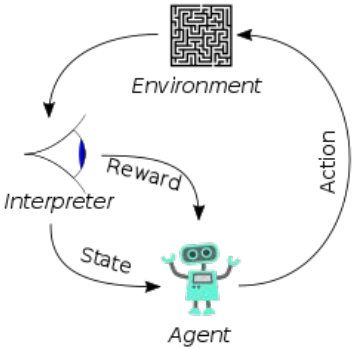
- Most ML models are trained like: $\text{minimize} \quad \text{loss}(f(x;\theta), y) + g(\theta)$
  - **x** is called... “input 入力”, “features 特徴量”, “explanation variable 説明変数”
  - **y**... “labels 正解ラベル”, “ground truth”, “gold”, “target variable 目的変数”
  - **(x, y)**... “dataset データセット”
  - **f(θ)**... This is the **machine**! With parameters (machine’s state) θ
  - loss... **Loss function** 損失関数
    - ex. Mean squared error, Cross entropy loss, etc...
  - **g(θ)**... Regularization terms
- Example: Item price prediction
  - x = (item’s name, category, brand)
  - y = price
  - f = linear regression model
  - loss = Mean squared logarithmic error

Ref)
https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf

ML Common Patterns

- Most ML models are trained like: $\text{minimize} \quad \text{loss}(f(x;\theta), y) + g(\theta)$
- **Supervised Learning**
  - Train model(s) so that the inference result is close to the target variable
  - ex. Predicting item price from given item information
  - ex. Detecting not appropriate messages
- **Unsupervised Learning**
  - Train model(s) without target variables (x ~= y)
  - ex. Creating item embedding using word2vec, ChatGPT*
- **Reinforcement Learning**
  - Train model(s) from reward given from environment
  - The model f(x) decides the action to the environment
  - ex. Mercari home screen optimization (Multi-Armed Bandit)
  - ex. AlphaGo, Auto-driving system
- etc

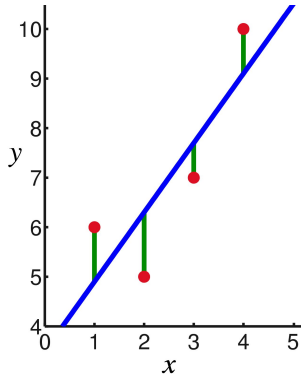




※Images are from wikipedia.com, Public domain or CC 0

Machine Learning Basics - Supervised Learning

- **Regression 回帰**
  - Target variable is normally continuous
    - ex. Item price, images, audio, etc.
  - Loss
    - MAE, MSE, LMSE, MSLE, etc.
  - ex. Predicting item price from given item information



- **Classification 分類**
  - Target variable is normally categorical
    - ex. Item category, spam or not, etc.
  - Loss
    - 0 or 1, logistic loss, cross entropy loss, etc.
      - Differentiable entropy from prob distribution to target label
  - ex. Detecting not appropriate messages

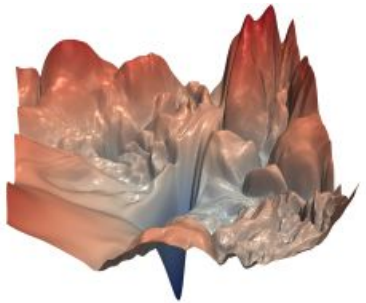
※Images are from wikipedia.com, Public domain or CC 0

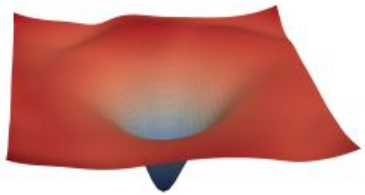
How do machines learn?

- Minimize Loss
  - Regression: Mean Squared Error
    - Measures how far your predicted value is from the actual value on average
  - Classification: Cross-Entropy
    - Measures how confident you are in your correct and incorrect predictions
- (Stochastic) Gradient Descent
  - Differentiate loss and go down
    - Local optima vs global optima
  - Designing and choosing appropriate loss functions is key to solving a ML problem

$$\mathrm{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2$$

$$L_{\log}(y, p) = -(y \log(p) + (1 - y) \log(1 - p))$$

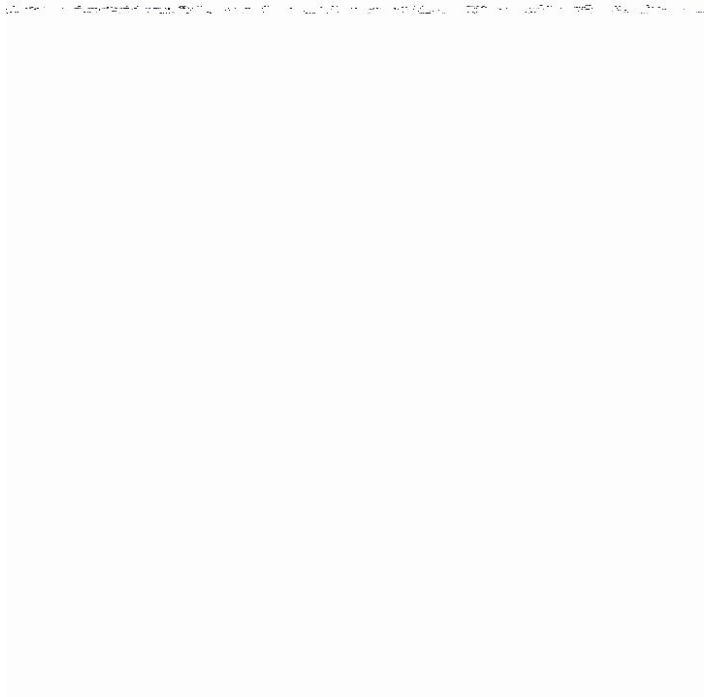




Ref)
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html
https://scikit-learn.org/stable/modules/model_evaluation.html#mean-squared-error

How do machines learn?

- Example: Linear Regression (y=wx+b)
  - Dataset: (x, y)
    - ex. Predicting penguin's height from weight
    - Two parameters: w and b
  - Using MSE
  - Differentiate:
    - $(wx+b - y)^2 / dw = 2x(wx+b - y)$
    - $(wx+b - y)^2 / db = 2(wx+b - y)$
  - Set any initial value for w and b
  - For each training batch:
    - $w \leftarrow w + \alpha 2x(wx+b - y)$
    - $b \leftarrow b + \alpha 2(wx+b - y)$
  - Here α is the learning rate
  - Same whether x is a scalar or a vector

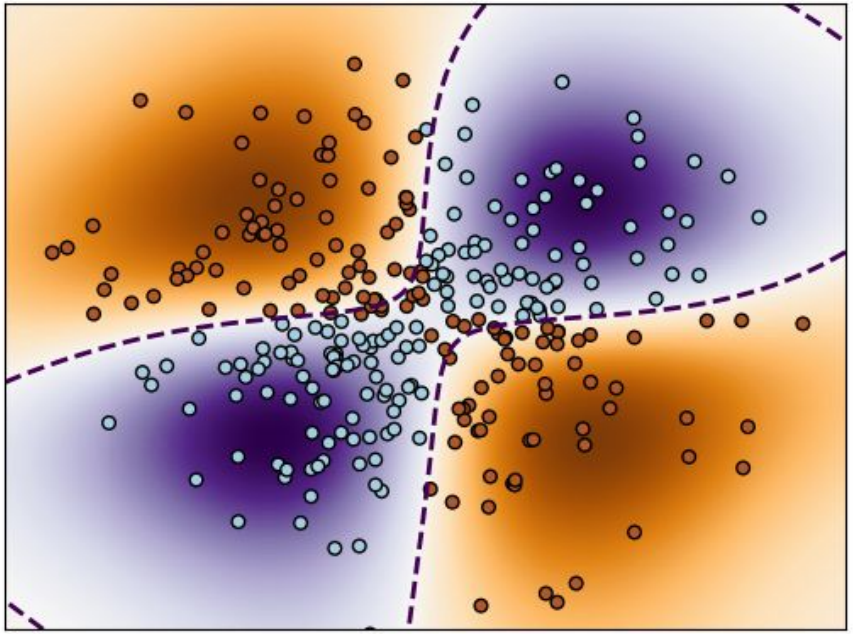


Ref)
https://ruder.io/optimizing-gradient-descent/
*https://towardsdatascience.com/gradient-descent-animation-1-simple-linear-regression-e49315b24672

mercari

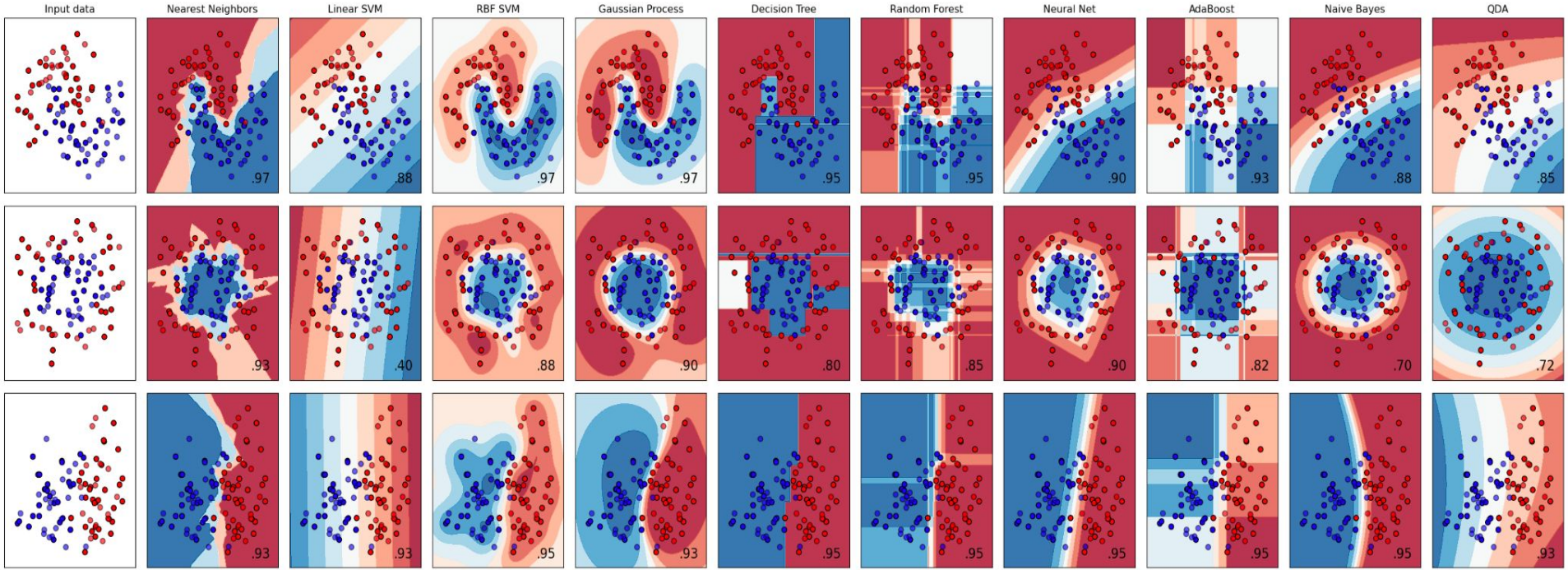
What if things do not seem linear?

- Just use non-linear machine
  - Kernel functions allow you to transform features into spaces where classes are linearly separable
- Non-linear models are complex but powerful
  - Support vector machine
  - Boosting trees
  - Neural networks
- But the principle is the same!



Ref)
https://gregorygundersen.com/blog/2019/12/10/kernel-trick/
https://scikit-learn.org/stable/auto_examples/exercises/plot_iris_exercise.html#sphx-glr-auto-examples-exercises-plot-iris-exercise-py

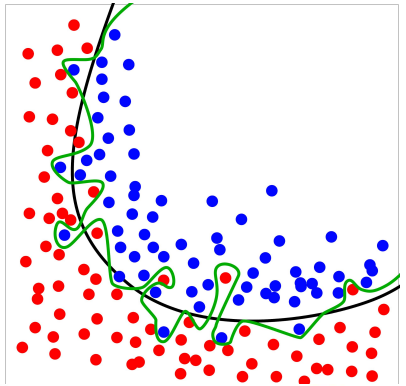
Linear/Non-Linear Models

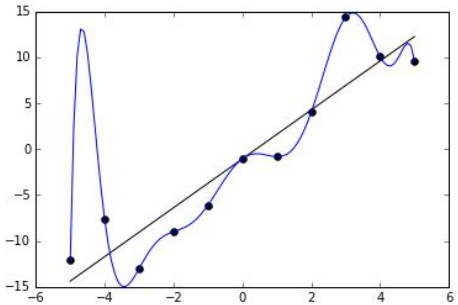


Ref)
https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html?highlight=comparison

Trade off: Underfitting vs Overfitting

- But should we use the most complex model and many features?
  - Ability to **generalize** is important!
  - “Training data” is not “all possible data”
  - Trade-off:
    - Fitting to training data
    - Robustness to new data
  - In other words: Bias vs Variance
- How to control the trade off?
  - **Dataset split** (ex. train/validation/test)
    - Training a model with train set
    - Stop training once the loss for vld set is increased
    - Evaluate a model performance with test set
  - **Ensemble model**
    - Using multiple model to single problem
  - etc





Ref)
https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html#sphx-glr-auto-examples-model-selection-plot-underfitting-overfitting-py

Other Trade Off

- **Speed vs Accuracy**
  - Large model is strong and slow
  - Depending on the project
    - Light model for real-time inference
    - High performance model for batch jobs
- **Cost vs Accuracy**
  - Advanced model, Ensemble model, Complex preprocessing...
  - Many costs
    - Inference cost, training cost, maintenance cost, onboarding cost...
  - Set (ML specific) SLO first
    - Target accuracy, maximum latency,
  - Stand on the shoulders of giants (use flameworks!)
    - Many papers on machine learning
    - Modeling tools (scikit-learn, Tensorflow, PyTorch...)
    - Training/Monitoring platform (Kubeflow, DataDog...)

Ref)
https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html#sphx-glr-auto-examples-model-selection-plot-underfitting-overfitting-py

Preprocessing

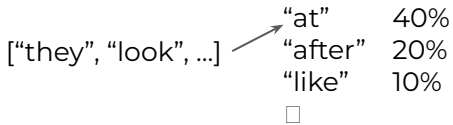
- How do we input data to machine?
  - Models can easily understand scalar, vector, matrix, tensor…
  - How about categorical data, text, audio or image?
    - Preprocessing!
- Example: One-hot encoding
  - Create a vector in which only one element has 1 and the others have 0
  - ex. The day of week: Monday → [0,1,0,0,0,0,0], Wednesday → [0,0,0,1,0,0,0]
- Example: Text and bug-of-words
  - Build dictionary and count words. Each word corresponds to defined element.
  - ex. “dog cat bird” → [1,1,1], “dog cat dog” → [2,1,0], “dog dog dog dog” → [4,0,0]
  - Now you can input any sentence as a vector!
- And more…
  - Data generation and preprocessing are most important parts of practical ML

mercari

Ref) https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf

More Examples

- Language Model
  - Language model is basically probability distribution for word sequence
  - Techniques/preprocessing
    - One-hot encoding for neural networks
    - N-gram (Treating consecutive words as one word)
      - ex. “Time fries arrow” → [“time fries”, “fries arrow”]
    - Markov modeling
  - Example: ChatGPT, Instruct GPT
    - Training language models to follow instructions with human feedback [Ouyang+, ‘22]
    - **175B** parameters! (with GPT-3)
      - The penguin model had only 2 parameters 😮‍💨
    - Supervised learning + Reinforcement learning

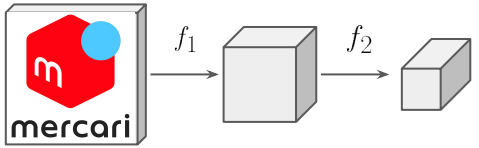


$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\text{RL}}}}\left[r_\theta(x,y) - \beta \log\left(\pi_\phi^{\text{RL}}(y \mid x)/\pi^{\text{SFT}}(y \mid x)\right)\right] + \gamma E_{x\sim D_{\text{pretrain}}}\left[\log(\pi_\phi^{\text{RL}}(x))\right] \quad (2)$$

Ref) https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf

More Examples

- Machine Learning for Images
  - Modern method is deep learning!
    - Process image as a three-dimensional tensor
    - Height*Width*Color (RGB)
  - Convolutional Neural Network(CNN)
    - Imitating human visual cortex
    - Convolve pixels using kernels
  - Legacy method: Hand-crafted feature extraction
    - Dimension reduction for generalization(PCA, SIFT, etc.)
    - Image is basically same even if a pixel is different
  - Example:
    - Image search
    - Semantic segmentation for auto driving
    - Blurred background



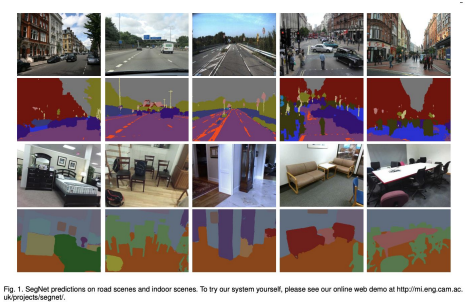


Fig. 1. SegNet predictions on road scenes and indoor scenes. To try our system yourself, please see our online web demo at http://mi.eng.cam.ac.uk/projects/segnet/.

[Badrinarayanan+, '16]

Ref) https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf

ML Practices

Considerations for building ML applications

- What ML is good at
  - Automating work that requires a lot of human effort
    - Human = customer (best case!), CS agents, etc
  - Collective Intelligence (集合知) approach
  - Hard if there's few data 😭
  - Advantages over statistics
    - Manual feature processing is not 100% necessary
    - The machine automatically select/combine features instead of you
- What ML is BAD
  - **High Cost**... Implementation cost, computer resource, maintenance cost...
- The more data, the better, but can we use all data points?
  - Data sampling, dirty data...
  - Data split for generalization performance check: Train, Validation, Test
  - Changing trends in data (Concept drift)
    - How do we deal with seasonal trends?

mercari

Ref) https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf

ML Project Lifecycle

- **ML project is HIGH COST** 🤯
  - Automation is not yet fully automated

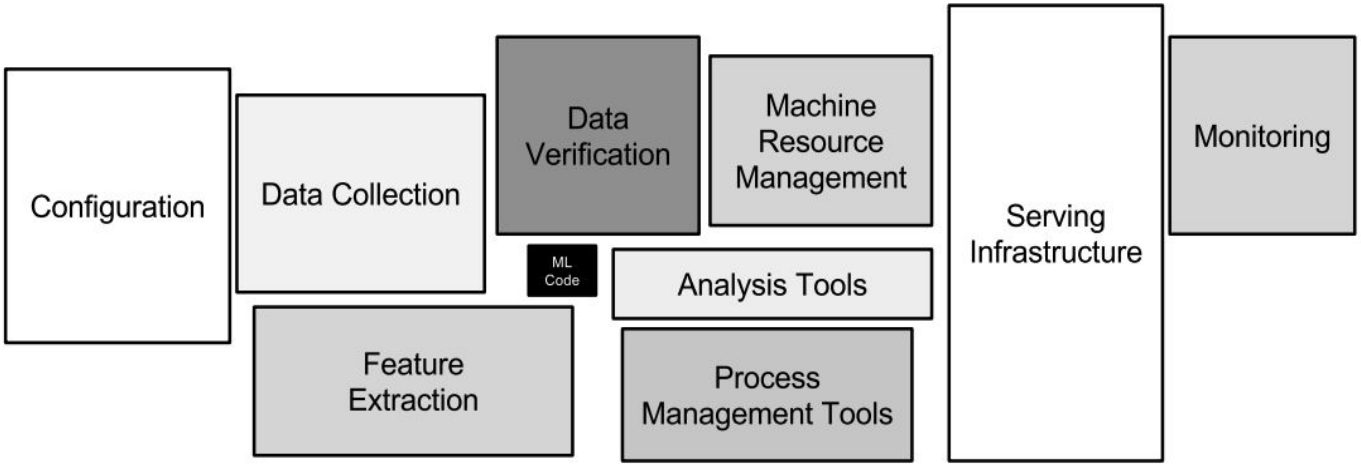
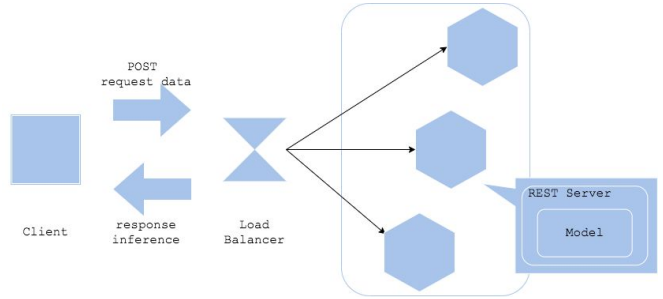


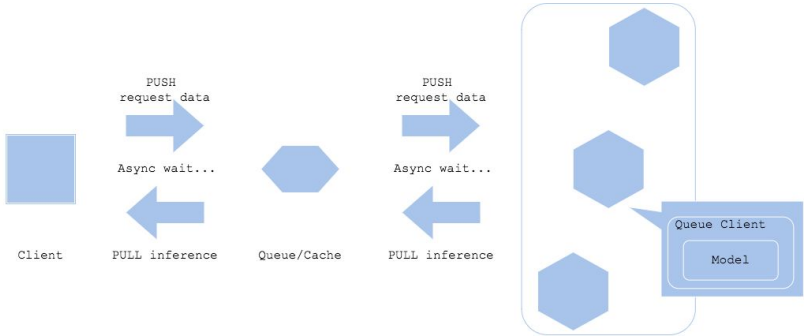
Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Reference:
https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html
https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html

ML Design Pattern

- Mercari publishes machine learning design patterns
  - Introduce typical serving/QA/monitoring patterns
  - Like GOF book
  - Example: Web single pattern
    - Simple
    - Each model have own server
  - Example: Asynchronous pattern
    - Asynchronously serve predictions
    - Not real-time but high availability



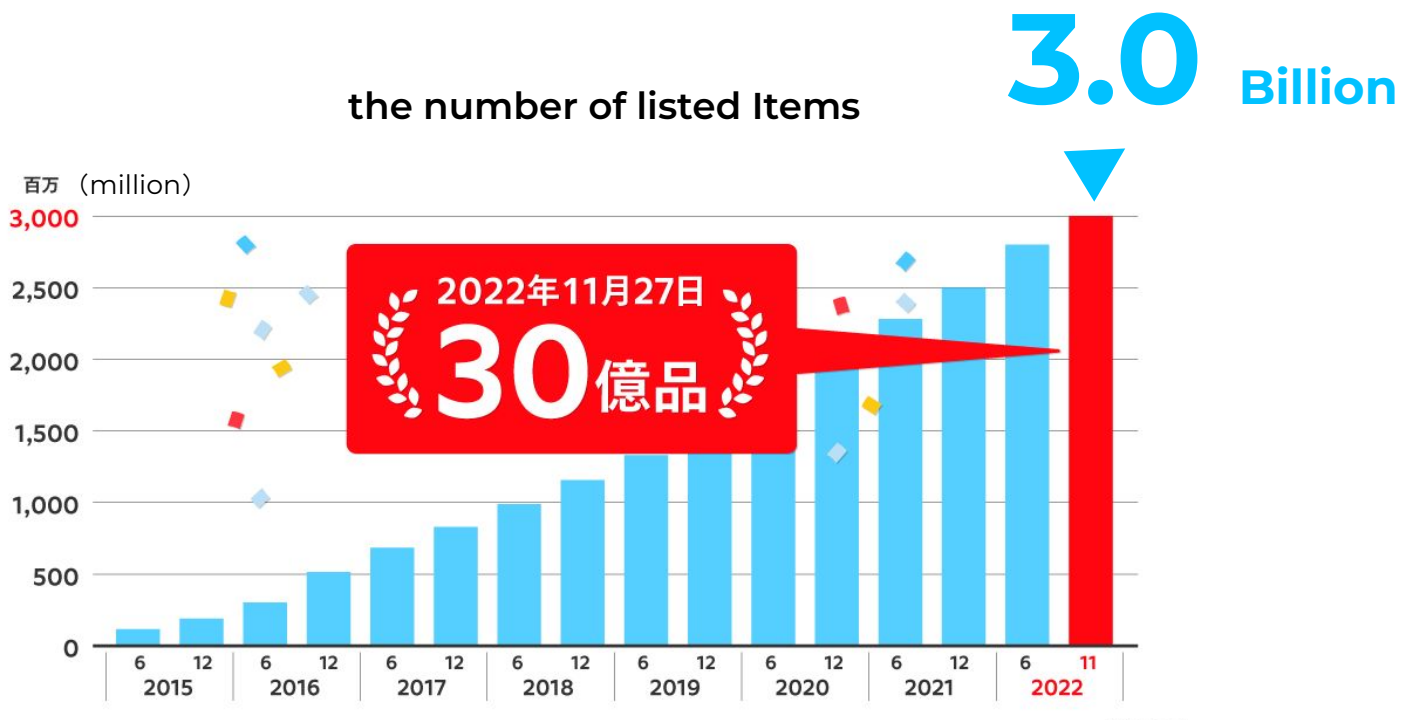


Ref)
https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html

Data at Mercari

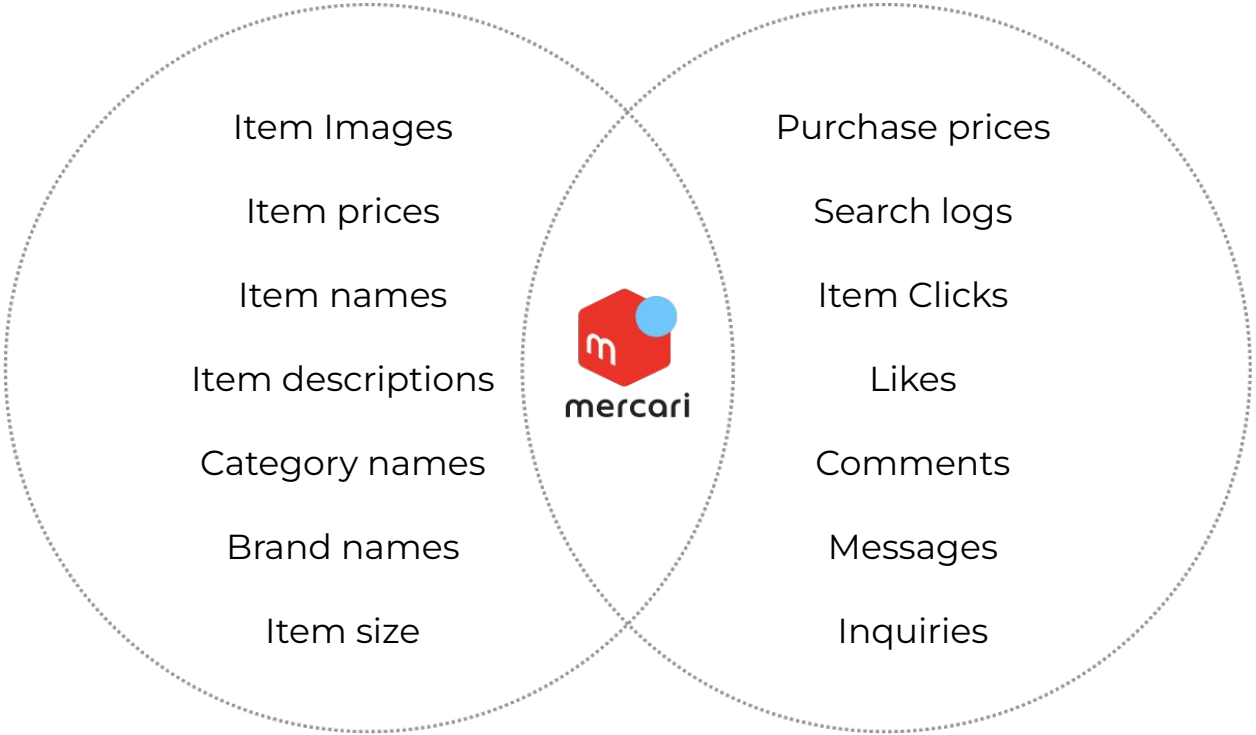
Large scale dataset

More than **3.0 billion items with image and text data**



Large scale dataset

Billions of listing and buying



ML Projects at Mercari JP

Disclaimer

- **Mercari tests many features quickly**
- **Content might be different from the latest version 🤦**

(part of) AI in Mercari

	2017-2018	2019-2020	2021-
Buy & Sell		**- Real-time recommend** - Coupon optimization	**- Layout personalization** **- Advanced SERP reranking** - Notification optimization
Listing	- Price suggestion v1 - AI & Barcode listing	- Price suggestion v2 - Catalog Automapping	- Metadata tagging
Safe	- Item moderation v1	- Item moderation v2 - Message moderation v1	- Message moderation v2
Platform	- ML Platform v1	- ML Platform v2 - Image search - Edge AI	- Customer support excellence

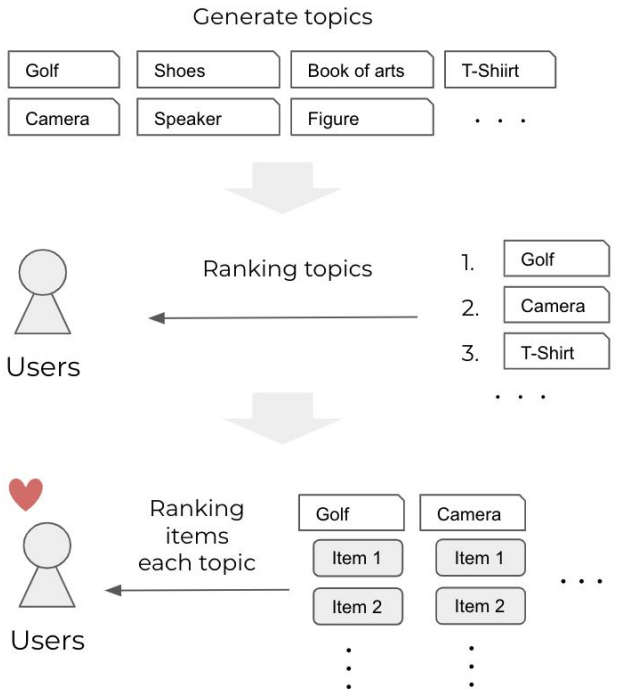
mercari

Basic Flow of Home Recommendation

1. **Create a topic**
   Clustering / labeling products with appropriate item cluster (The substance as a system is a search filtering condition)

2. **Rank topics**
   Provide appropriate topics based on user behavior history, etc.

3. **Rank products within the topic**
   Rank products based on user and product data



Realtime Retargeting

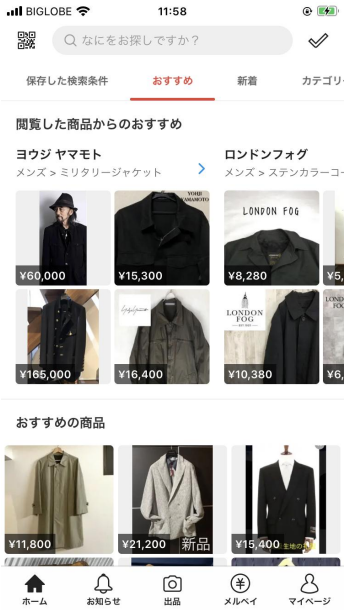
New component on Home screen for recommended items

- Show explainable recommendations based on customers' recent browsing history:
  - Pick up keyword category pair or brand category pair based on recent activity, and display items plus entrance to search from these items.
  - Each pair is generated by recent users' browsed items, with a weighting system that puts more weight on most recent activity.
  - Contents of component is changing in real time following user's browsing behaviour; if customer views a new items, recommendation is updated as soon as customer comes back to Home screen.

Layout Optimization

Personalization of Home Components

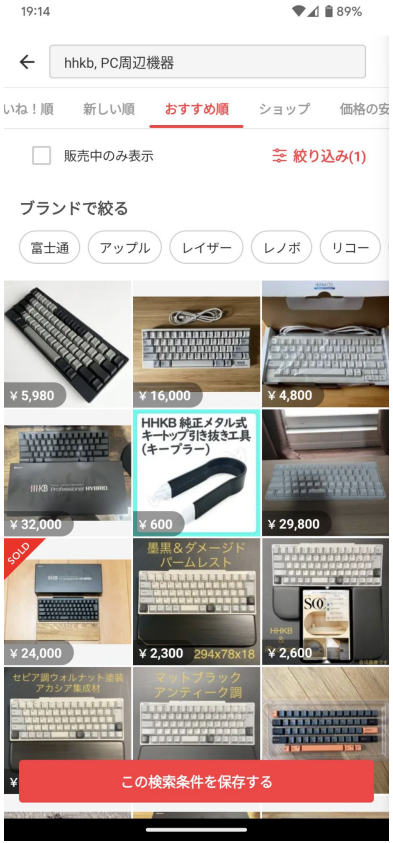
- We have some components for home screen
  - Recommendation from viewed/liked item
  - Simply showing viewed/liked item
  - And more
- We optimize the order of components
  - In addition to the content of each component
- Using Multi-armed bandit (MAB)
  - Kind of **reinforcement learning**!



Advanced SERP reranking

Long Journey to Machine-Learned Re-ranking

- SERP = Search Engine Result Page
  - Large amount of transactions starts from here!
  - Mercari blog [@alex, ‘21]
- Learning-to-Rank
  - ML scheme to rank items based on user preference
  - Basically supervised learning
- Many challenges
  - Data labeling (data collection)
  - Position bias
  - User context
  - Contribution to business metrics
  - etc.



(part of) AI in Mercari

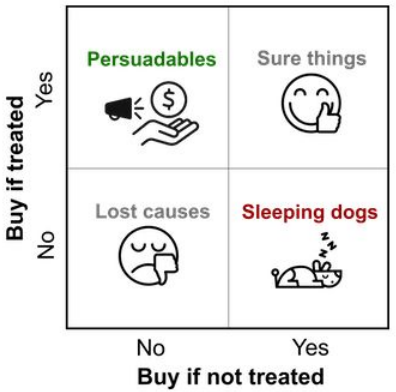
	2017-2018	2019-2020	2021-
Buy & Sell		- Real-time recommend **- Coupon optimization**	- Layout personalization - Advanced SERP reranking - Notification optimization
Listing	- Price suggestion v1 - AI & Barcode listing	- Price suggestion v2 - Catalog Automapping	- Metadata tagging
Safe	- Item moderation v1	- Item moderation v2 - Message moderation v1	- Message moderation v2
Platform	- ML Platform v1	- ML Platform v2 - Image search - Edge AI	- Customer support excellence

mercari

Data Driven Marketing utilizing ML

Utilize ML to promote data driven marketing campaigns

- Project Examples
  - Buyer coupon distribution optimization
    - Remove organic users (sure things)
      - Predict who will buy without a coupon
      - Achieved a cost reduction effect of nearly 50 million yen per year by suppressing unnecessary coupon distribution

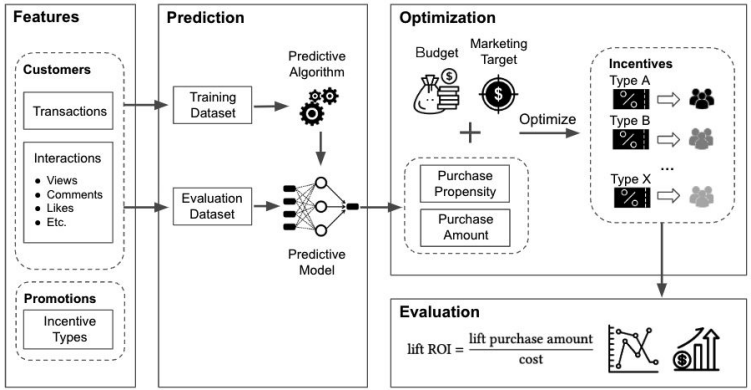


mercari

Data Driven Marketing utilizing ML

Utilize ML to promote data driven marketing campaigns

- Project Examples
  - Buyer coupon distribution optimization
    - Optimizing incentive amount for each user
      - Using uplift-modeling + mathematical optimization to further optimize coupon distribution target selection



(part of) AI in Mercari

	2017-2018	2019-2020	2021-
Buy & Sell		- Real-time recommend - Coupon optimization	- Layout personalization - Advanced SERP reranking - Notification optimization
Listing	- Price suggestion v1 **- AI & Barcode listing**	- Price suggestion v2 **- Catalog Automapping**	- Metadata tagging
Safe	- Item moderation v1	- Item moderation v2 - Message moderation v1	- Message moderation v2
Platform	- ML Platform v1	- ML Platform v2 - Image search - Edge AI	- Customer support excellence

mercari

Goal of listing

Make listing as easy as possible

Just by taking a photo of item or barcode,

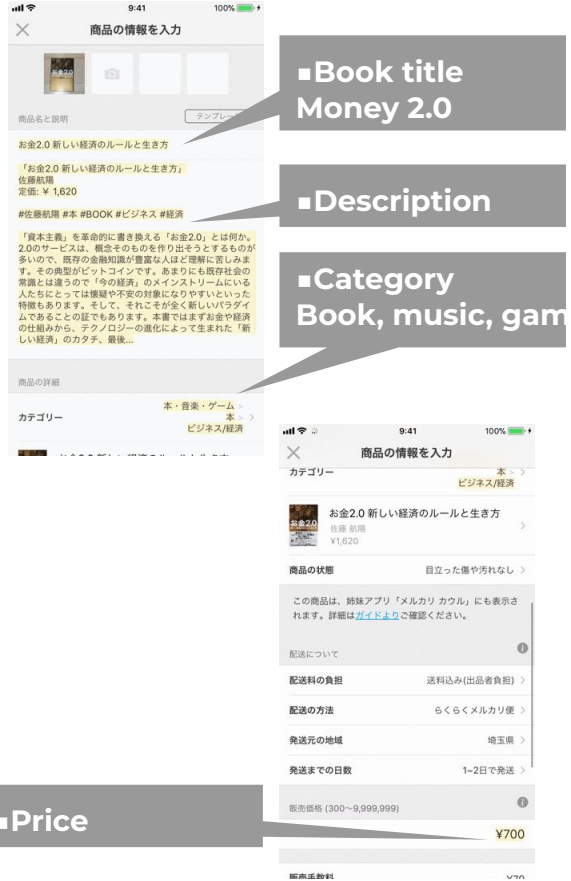
make it possible to list with one button

AI listing & Barcode listing

Photo

Barcode

Book, game, CD, cosmetics, etc



mercari

Barcode listing

AI listing

Fill out item title, description, category and brand based on image

Evolution of AI listing

AI in Mercari

	2017-2018	2019-2020	2021-
Buy & Sell		- Real-time recommend - Coupon optimization	- Layout personalization - Notification optimization
Listing	- Price suggestion v1 - AI & Barcode listing	- Price suggestion v2 - Catalog Automapping	- Metadata tagging
Safe	- Item moderation v1	- Item moderation v2 **- Message moderation v1**	**- Message moderation v2**
Platform	- ML Platform v1	- ML Platform v2 - Image search - Edge AI	- Customer support excellence

mercari

Text Moderation for Trust and Safety (TnS)

- Transaction message monitoring
  - Textual Content Moderation in C2C Marketplace [Shido+, ‘22]
- Problem of Rule-based Monitoring
  - Low accuracy! Only few positive escalations over 100 messages checked by CS agents

S **“Sorry, the price is really too low. Is it possible for us to …”**

B **“To finish the deal at twitter and ditch the transaction fee?”**

S **“Exactly. If it’s okay, please follow my twitter @hogefugapiyo .”**

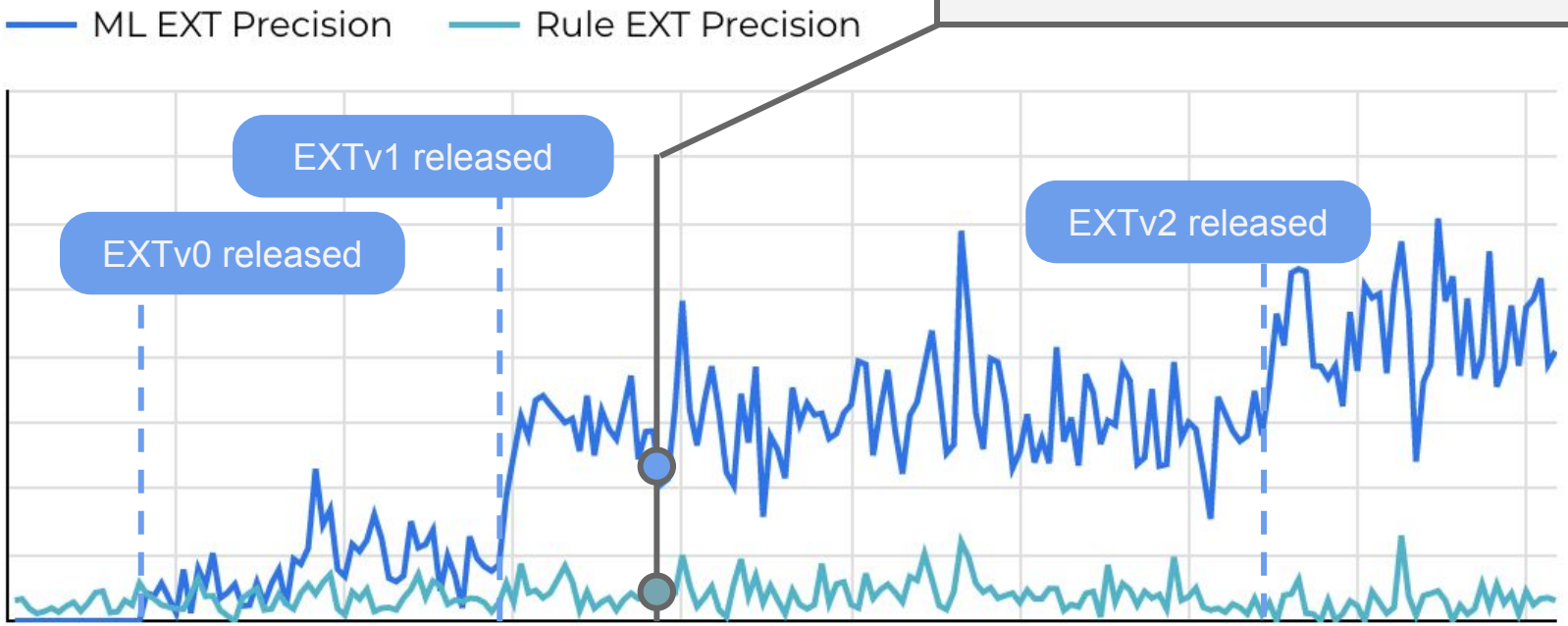
B **“Okay. Got it.”**

mercari

Online Evaluation

- EXT is a type of the violation

Rule patterns reported **twice the amount of ML reported alerts** but with merely **⅛ accuracy of ML-driven approach.**



mercari

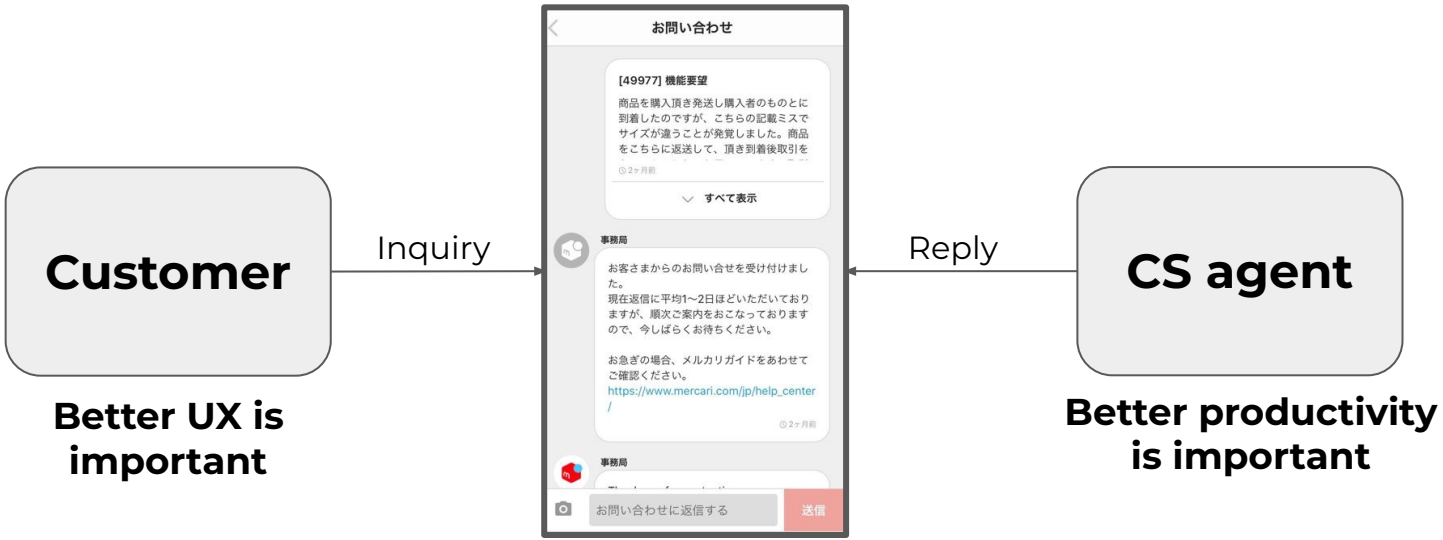
AI in Mercari

	2017-2018	2019-2020	2021-
Buy & Sell		- Real-time recommend - Coupon optimization	- Layout personalization - Notification optimization
Listing	- Price suggestion v1 - AI & Barcode listing	- Price suggestion v2 - Catalog Automapping	- Metadata tagging
Safe	- Item moderation v1	- Item moderation v2 - Message moderation v1	- Message moderation v2
Platform	- ML Platform v1	- ML Platform v2 - Image search - Edge AI	**- Customer support excellence**

mercari

Overview and Goals

Mission: Improve contact center operations & UX of inquiry with technology



Better UX is important

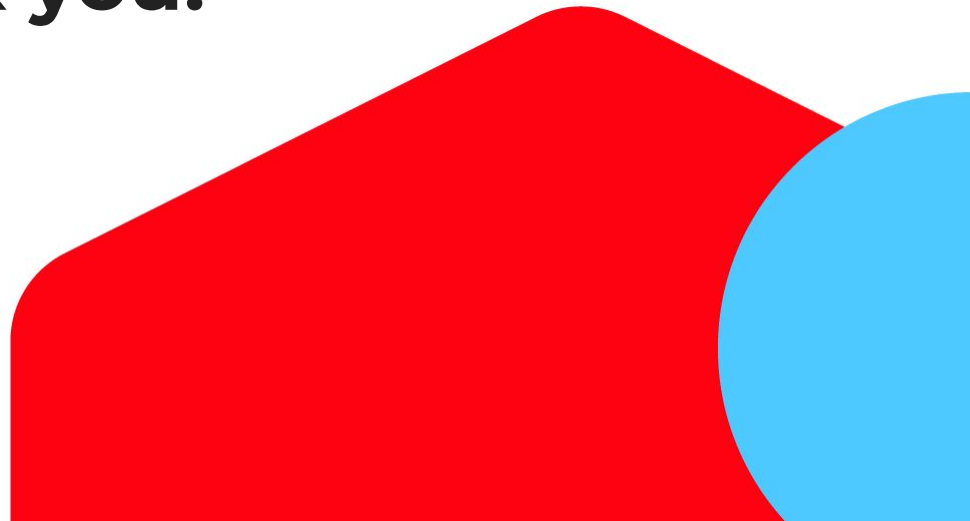
Better productivity is important

Chat-like Contact UI

Template suggestion for the contact tool

- 📖 What it is
  - Provide suggestions to CS agents in selecting the template to reply to customer inquiries.
- 🎯 Goal
  - It will reduce “Average Handling Time (平均対応時間)” of CS agents.

Thank you!



mercari