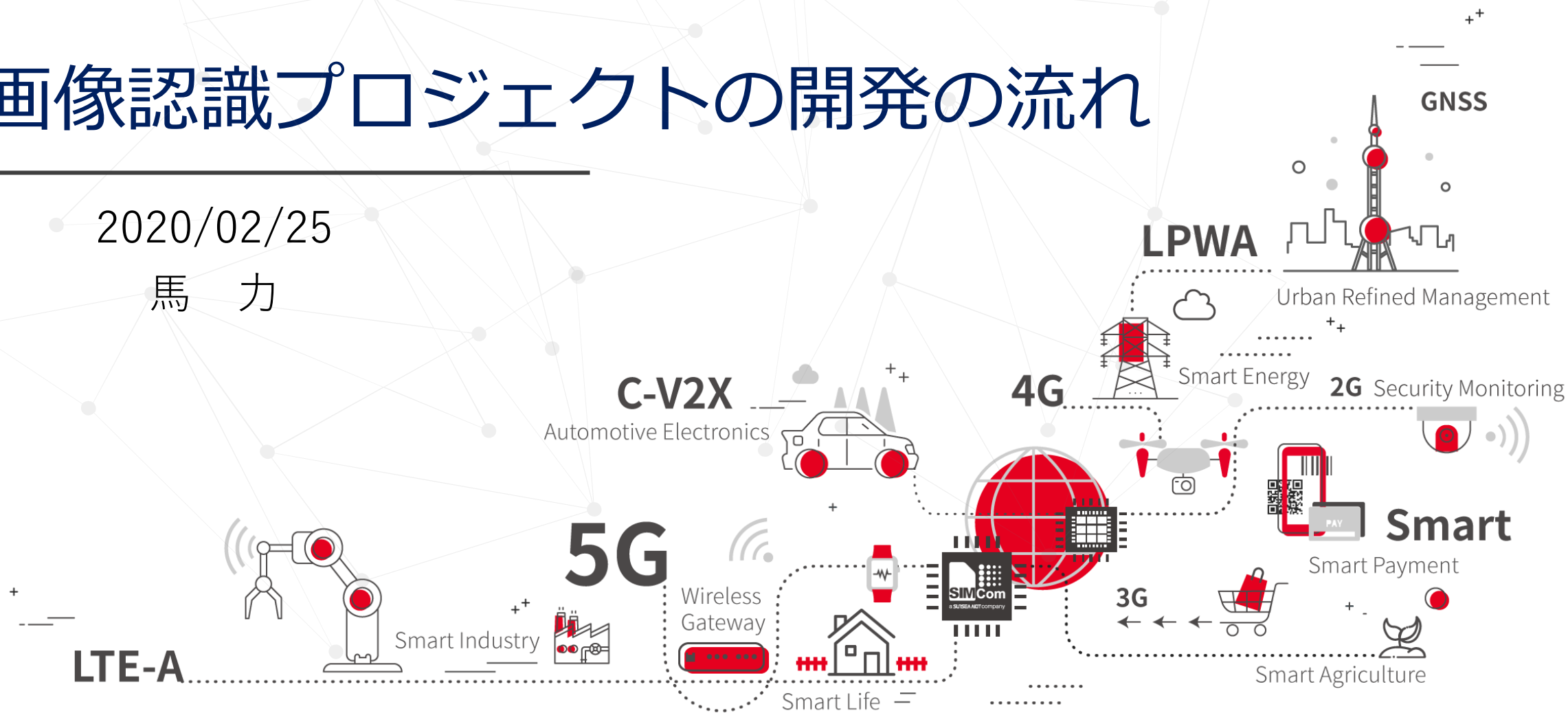


画像認識プロジェクトの開発の流れ

2020/02/25

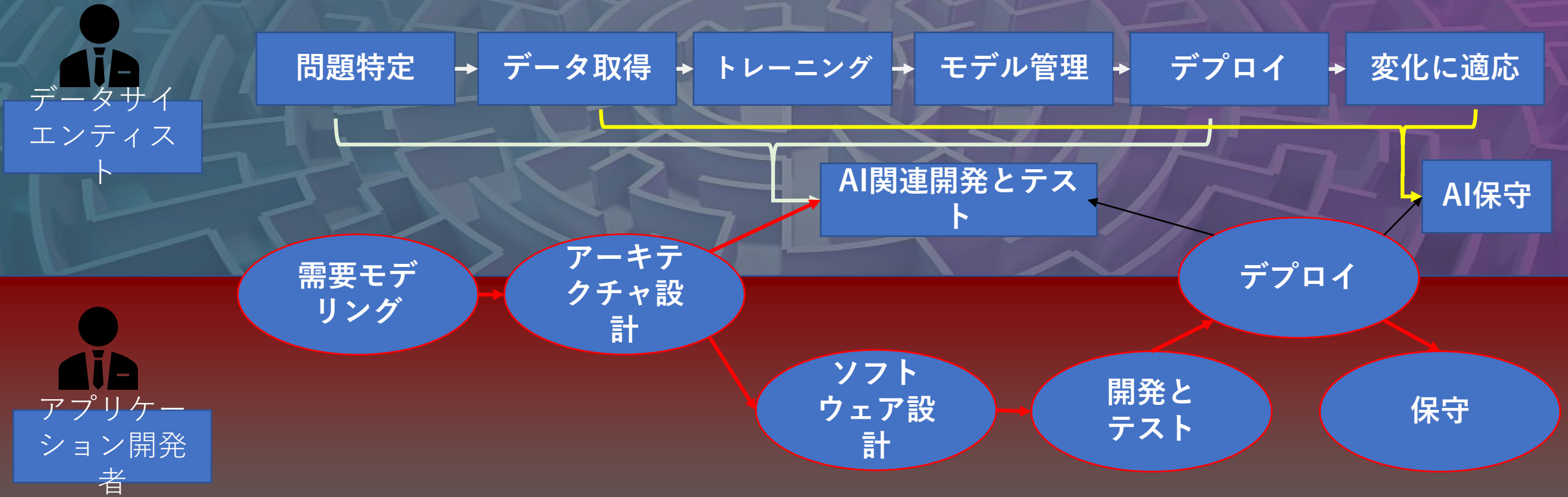
馬 力



開発方式

AI is redefining application development

AIはアプリケーション開発を再定義



問題特定

問題特定

Classification



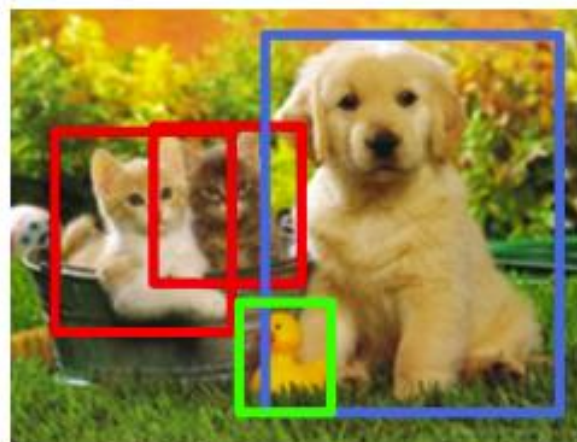
CAT

**Classification
+ Localization**



CAT

Object Detection



CAT, DOG, DUCK

**Instance
Segmentation**



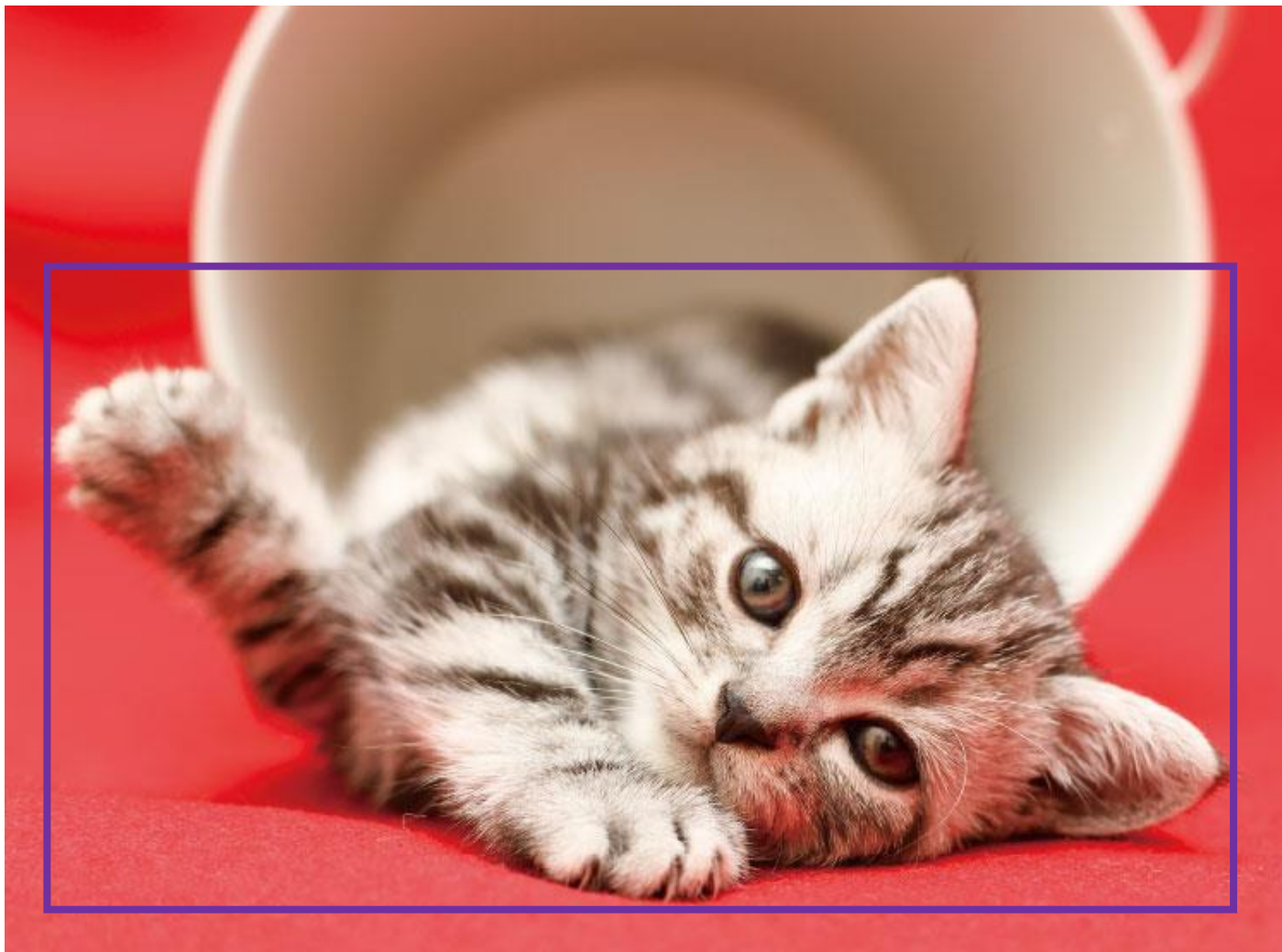
CAT, DOG, DUCK

データ準備

データ取得

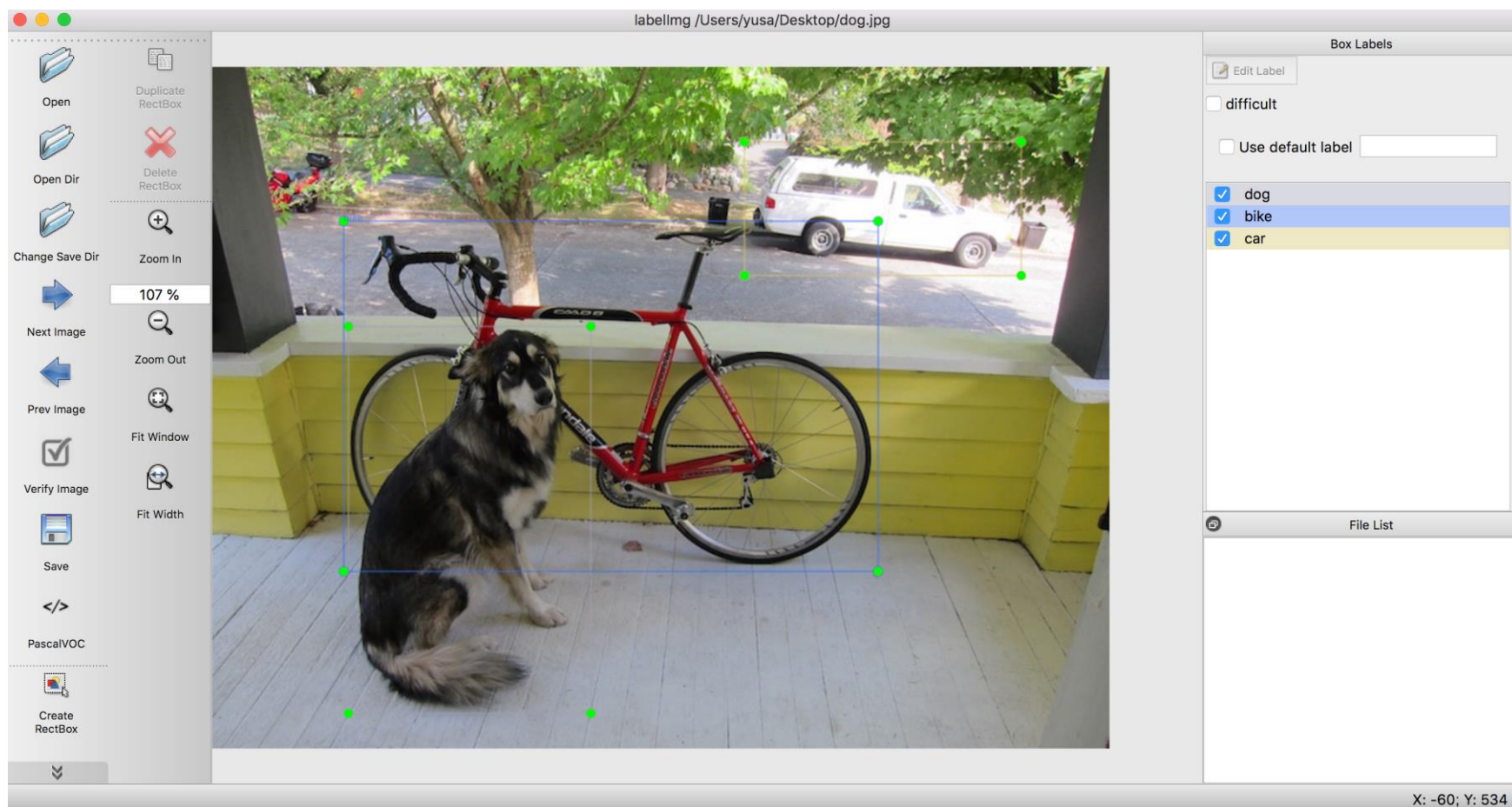


ラベリング



ラベリングツール labellmg

- **アプリタイプ:**
コマンド起動またはDesktopアプリ
- **LICENSE:**
MIT
- **入力フォーマット:**
ローカルファイル
- **出力フォーマット:**
PascalVOC
YOLOフォーマット



アルゴリズムの選択

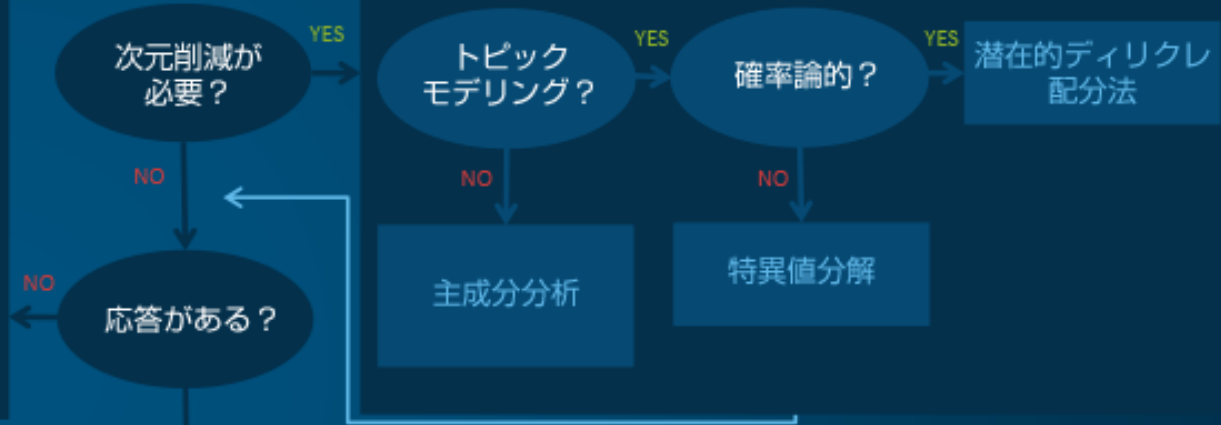
機械学習アルゴリズム選択チートシート

教師なし学習: クラスタリング

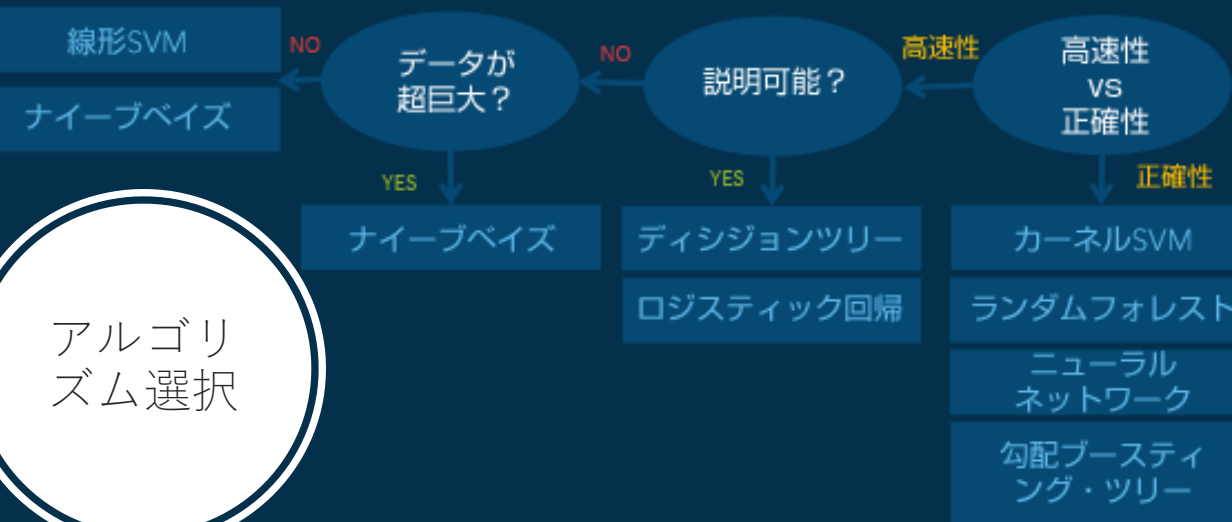


教師なし学習: 次元削減

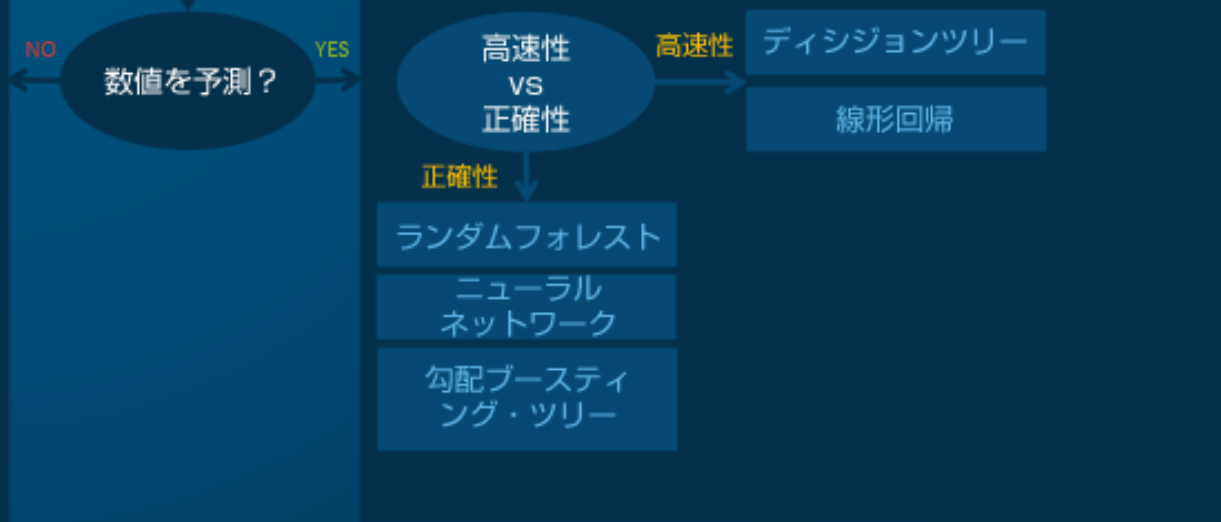
スタート



教師あり学習: 分類




教師あり学習: 回帰



アルゴリズム
選択

構成部分

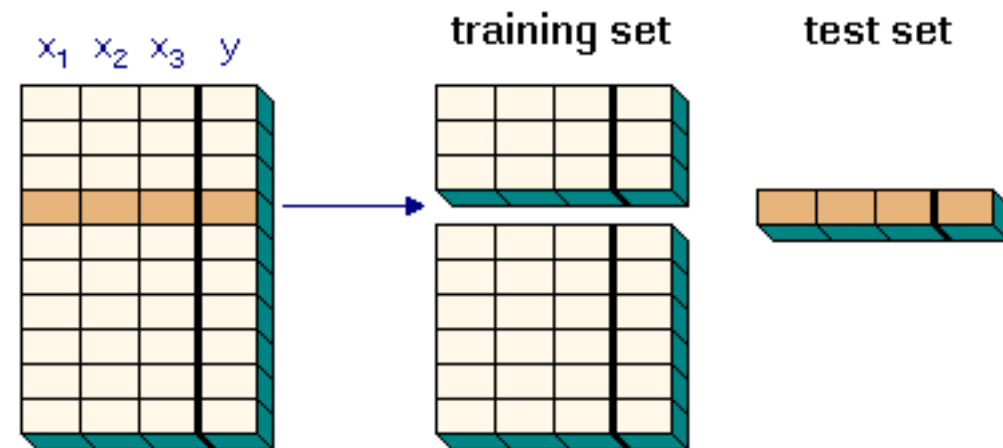
- ネットワーク構造
- 入力
- 出力
- パラメーター初期化
- 活性化関数
- 損失関数



トレーニング

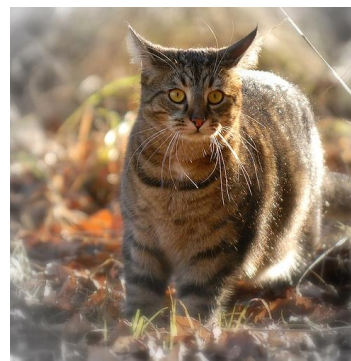
データの定義

- トレーニングデータ（キャリブレーションデータ）
 - 回帰モデル・クラス分類モデルの構築に用いるデータ
 - 目的変数の値・ラベルは分かっている
- バリデーションデータ
 - 回帰モデル・クラス分類モデルの検証に用いるデータ
 - 実際には目的変数の値・ラベルは分かっているが、わからないものとして（目隠し・ブラインドして）モデルから推定し、実際と推定結果とがどれくらいあっているか確認する
 - バリデーションデータで、モデルのハイパーパラメータ（PLSの最適成分数など）を最適化する
- テストデータ
 - 最終的にモデルの優劣を比較する



データ 水増し (data augmentation)

- 水平・垂直に画像をシフトする
- 水平方向・垂直方向に画像を反転させる
- 回転させる（回転角度はランダムの場合もある）
- 明度を変える
- ズームインする・ズームアウトする
- 画像の一部をくり抜く、削除する
- 背景色を変える
- 背景を置き換える
- Mixup・CutMix



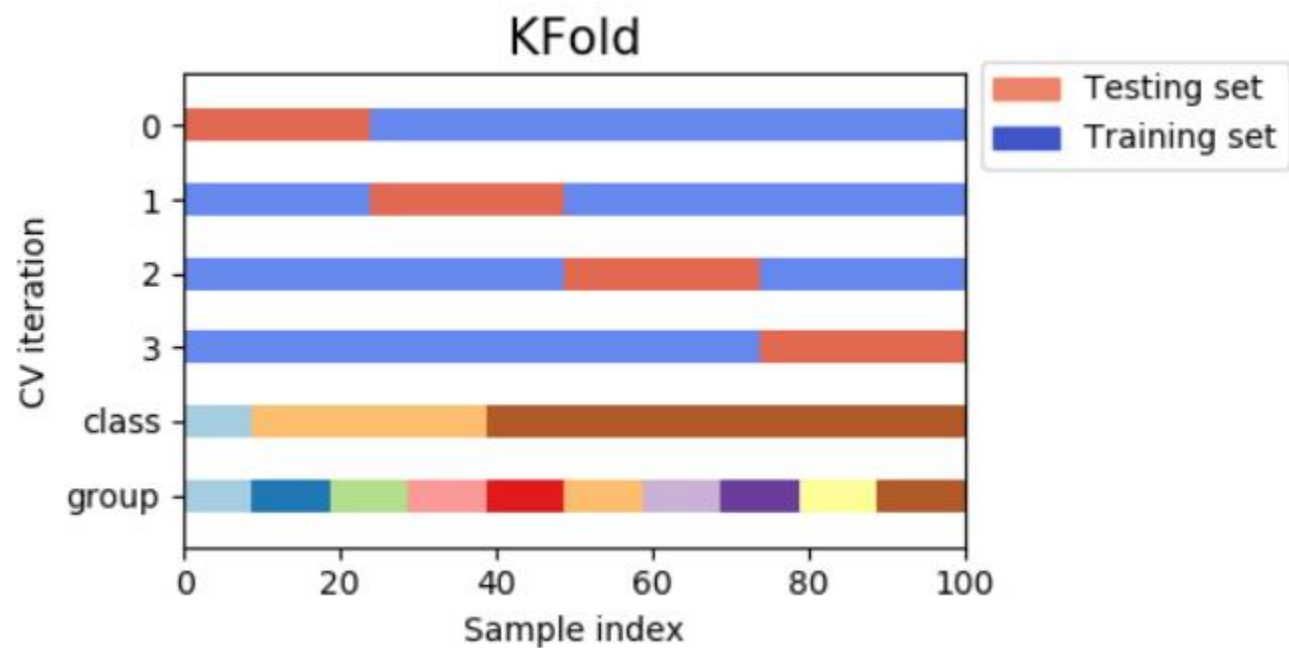
交差検証

- k分割交差検証 (k-fold cross-validation)
- 一つ抜き交差検証 (leave-one-out cross-validation)
- 層化k分割交差検証 (stratified k-fold cross-validation)
- シャッフル分割交差検証 (shuffle-split cross-validation)
- 層化シャッフル分割交差検証 (stratified-shuffle-split cross-validation)
- グループk分割交差検証 (group k-fold cross-validation)
- 時系列分割交差検証 (time series split cross-validation)
- スライド型の交差検証 (cross validation)

過学習を防いで汎化性能を高めるため

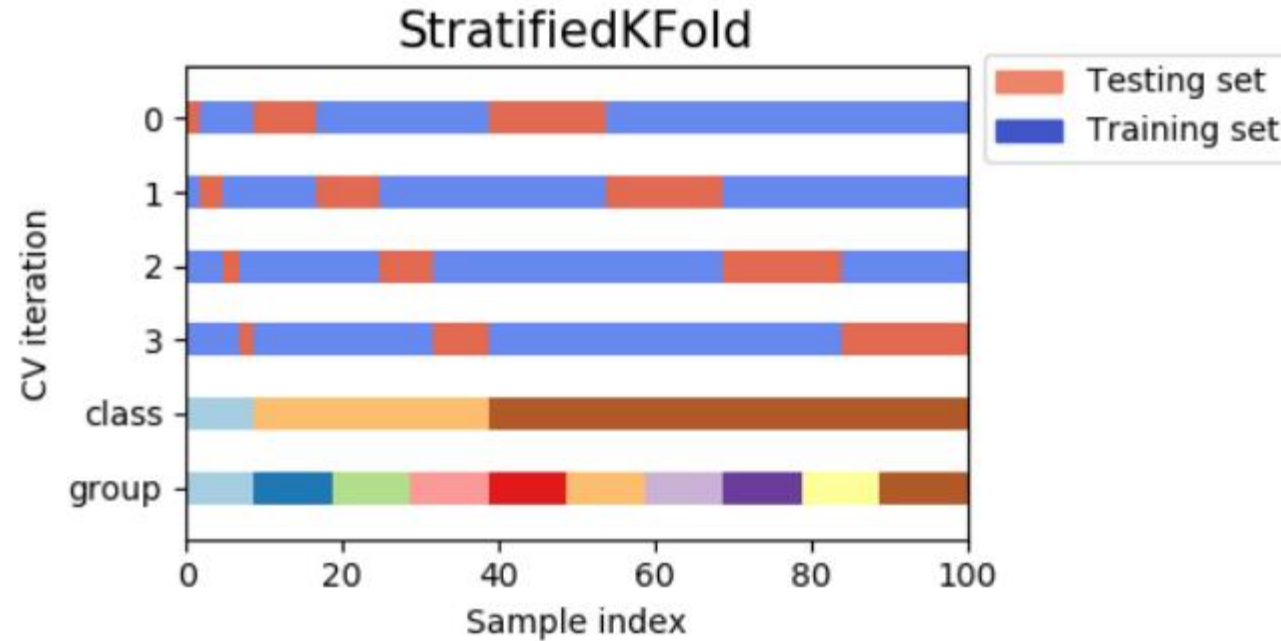
交差検証

k分割交差検証



交差検証

層化k分割交差検証 (stratified k-fold cross-validation)



交差検証

シャッフル分割交差検証 (shuffle-split cross-validation)



パラメータチューニング

- Learning rate
- Batch size
- Optimizer
- Iteration Times
- Epoch
- Early Stop

モデル性能評価

混同行列

		予測されたクラス	
		1 (Positive, 陽性)	-1 (Negative, 陰性)
実際の クラス	1 (Positive, 陽性)	True Positive (TP)	False Negative (FN)
	-1 (Negative, 陰性)	False Positive (FP)	True Negative (TN)




$$\text{正解率} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\text{検出率} = \frac{TP}{TP + FN}$$

$$\text{精度} = \frac{TP}{TP + FP}$$

$$\text{誤検出率} = \frac{FP}{FP + TN}$$

例

<div>ネコ！ (Predict : YES)</div> <div></div> <div>ネコ以外！ (Predict : NO)</div>			
ネコ (Actual : YES) 	90	10	合計 : 100
ネコ以外 (Actual : NO) 	20	30	合計 : 50
	合計 : 110	合計 : 40	合計 : 150

パターン

- 真陽性(True Positive)

「ネコの画像(Positive)を正しく(True)ネコと判別した」数で、左上の「90」

- 真陰性(True Negative)

「ネコ以外の画像(Negative)を正しく(True)ネコ以外と判別した」数で、右下の「30」

- 偽陽性(False Positive)

「ネコ以外の画像(Negative)を間違えて(False)ネコと判別した」数で、左下の「20」

- 偽陰性(False Negative)

「ネコの画像(Positive)を間違えて(False)ネコ以外と判別した」数で、右上の「10」

正解率(Accuracy)

- 全データのうち、正解したデータ数の割合




	ネコ! (Predict : YES)	ネコ以外! (Predict : NO)
ネコ (Actual : YES)	90	10
ネコ以外 (Actual : NO)	20	30

正解データ：
 $90 + 30 = 120$

正解率：
 $120 / 150 = 80.0\%$

再現率(Recall)

- 判別したいデータの数のうち、実際に判別できた割合
- 別名「**感度(Sensitivity)**」や「**検出力(Power)**」とも呼ばれる

	 ネコ！ (Predict : YES) ネコ以外！ (Predict : NO)	
ネコ (Actual : YES) 	90	10
ネコ以外 (Actual : NO) 	20	30

猫データに、AIの判断：
正解：90
不正解：10

再現率：
 $90 / (90 + 10) = 90.0\%$

特異度(Specificity)

- 判別したくないデータの数のうち、実際に判別されなかった割合

	<div>ネコ！ (Predict : YES)</div>	<div>ネコ以外！ (Predict : NO)</div>
<div>ネコ (Actual : YES)</div>	90	10
<div>ネコ以外 (Actual : NO)</div>	20	30

猫以外データに、AIの判断：
不正解：20
正解：30

特異度：
 $30 / (20 + 30) = 40.0\%$

適合率(Precision)

- “判別対象のデータ”を判別した数のうち、それが正解している割合、「陽性適中率(Positive Predictive Value)」とも呼ばれる



	<div>ネコ！ (Predict : YES)</div>	<div>ネコ以外！ (Predict : NO)</div>
<div>ネコ (Actual : YES)</div>	90	10
<div>ネコ以外 (Actual : NO)</div>	20	30

AIは猫データの判断結果に：
猫：90
猫以外：20

適合率：
 $90 / (90 + 20) = 81.8\%$

陰性適中率(Negative Predictive Value)

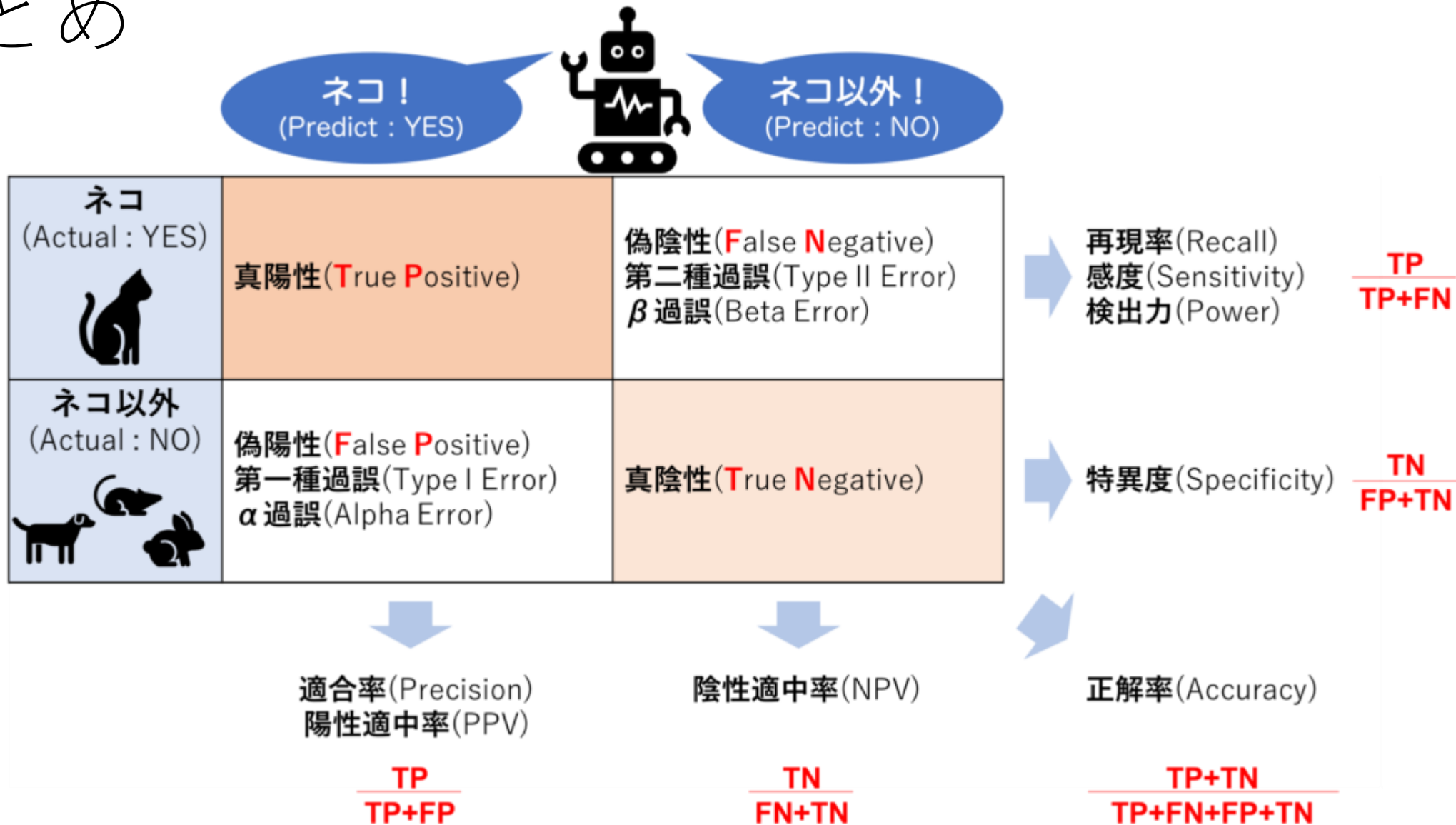
- “判別対象外のデータ”を判別した数のうち、それが正解している割合

	<div>ネコ！ (Predict : YES)</div>	<div>ネコ以外！ (Predict : NO)</div>
<div>ネコ (Actual : YES)</div> 	90	10
<div>ネコ以外 (Actual : NO)</div> 	20	30

AIから猫データ以外の判断結果に：
猫：10
猫以外：30

陰性適中率：
 $30 / (30 + 10) = 75.0\%$

まとめ



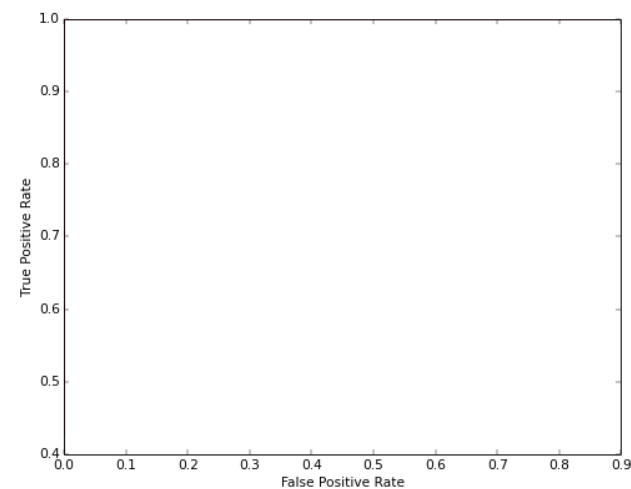
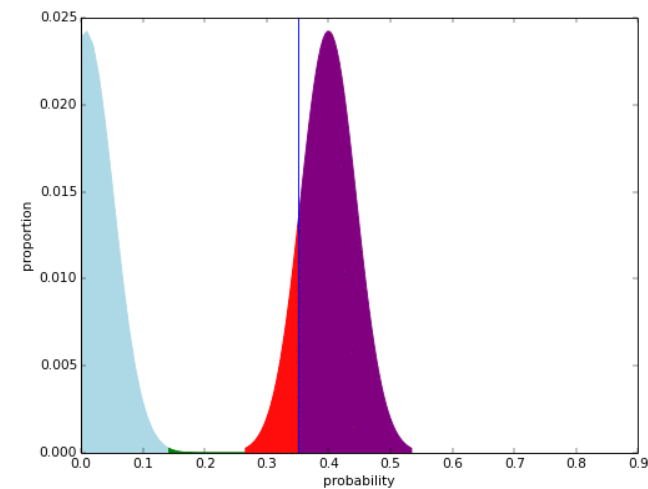
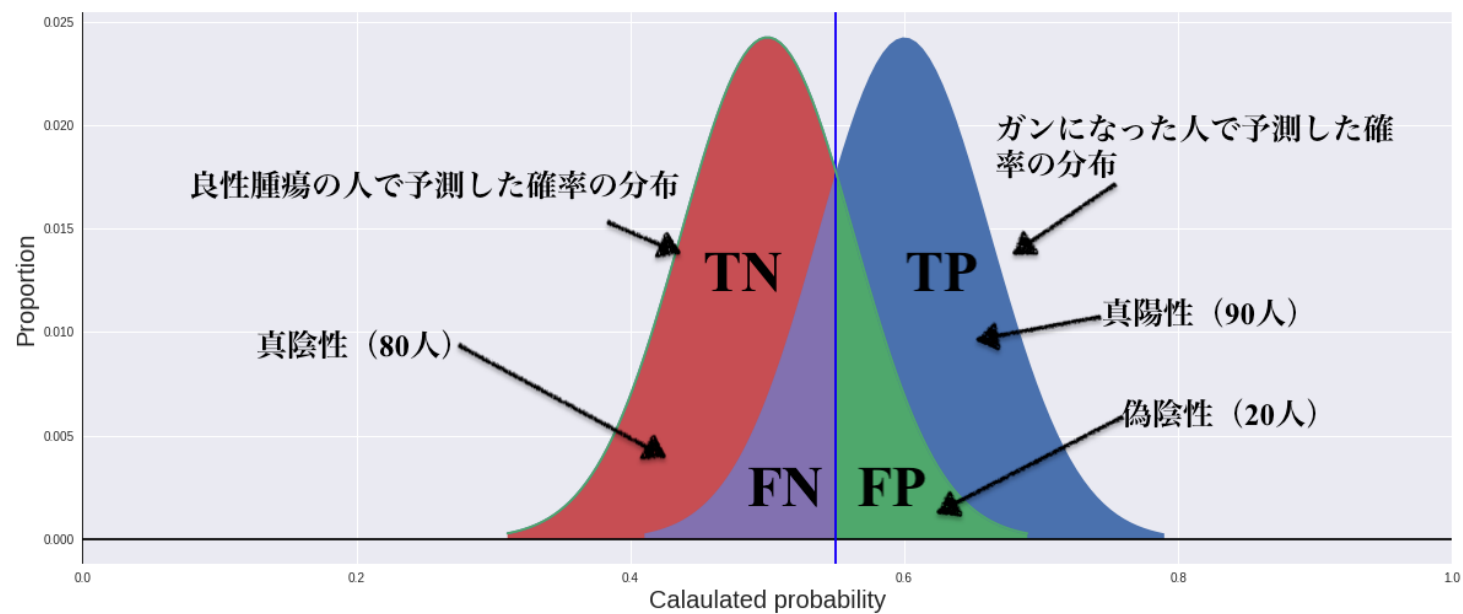
組み合わせによる指標

- F値(F-measure)

再現率と適合率の調和平均

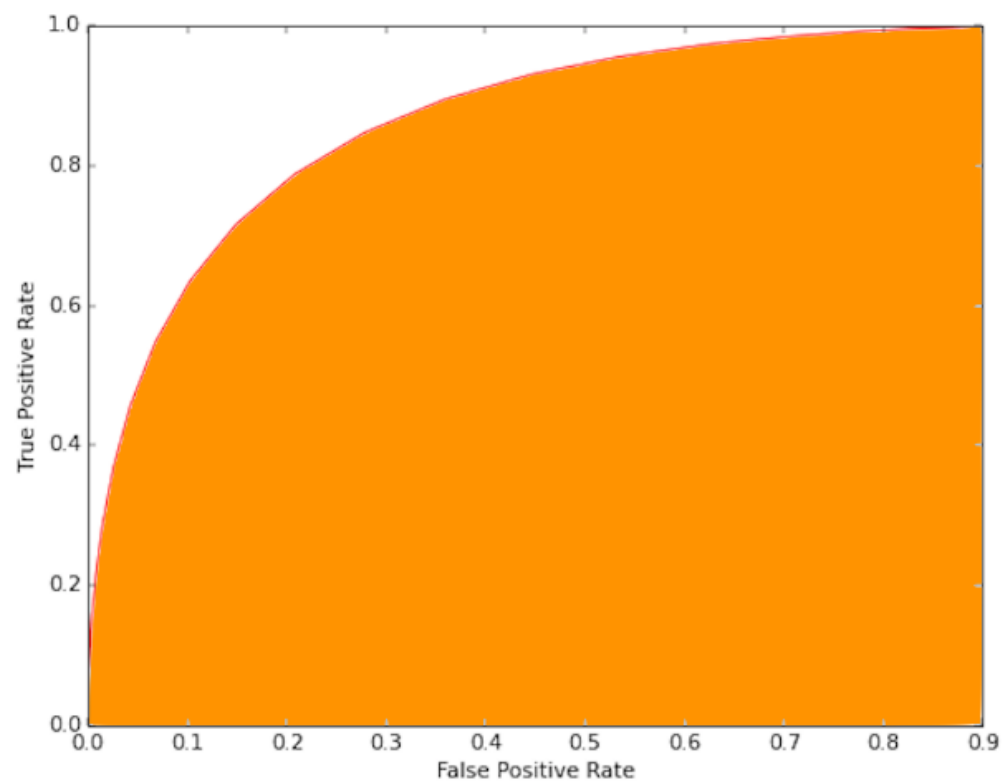
$$\frac{2\textit{Recall} * \textit{Precision}}{\textit{Recall} + \textit{Precision}}$$

ROC曲線

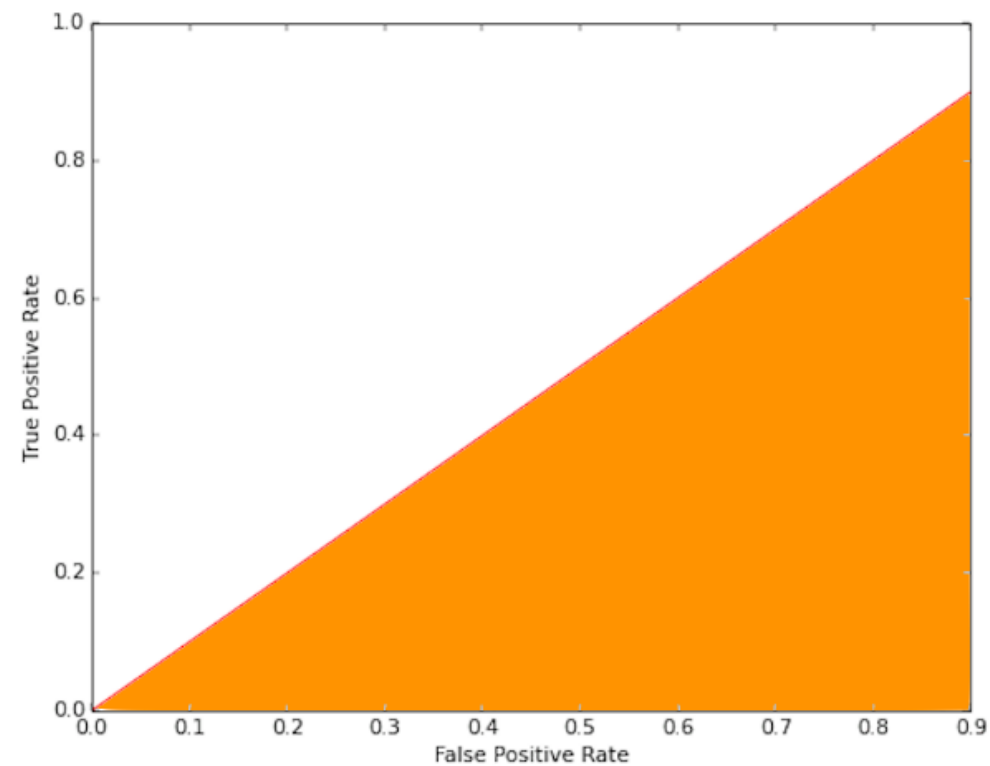


AUC

- ROC 曲線下の面積（積分）



効果がある分類器



効果がない分類器

AUC

- 一般範囲

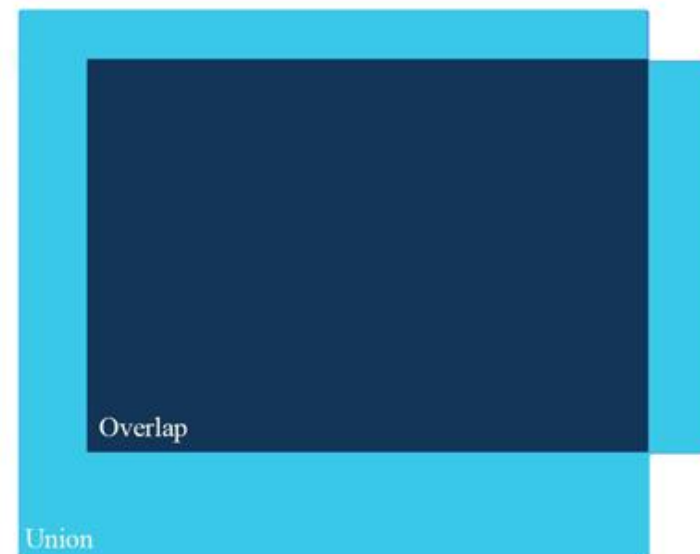
AUC	効果の評価
0.8以上	効果がすごくいい
0.7-0.8	効果があります
0.6-0.7	効果が多少あります
0.5-0.6	効果がありません

IoU

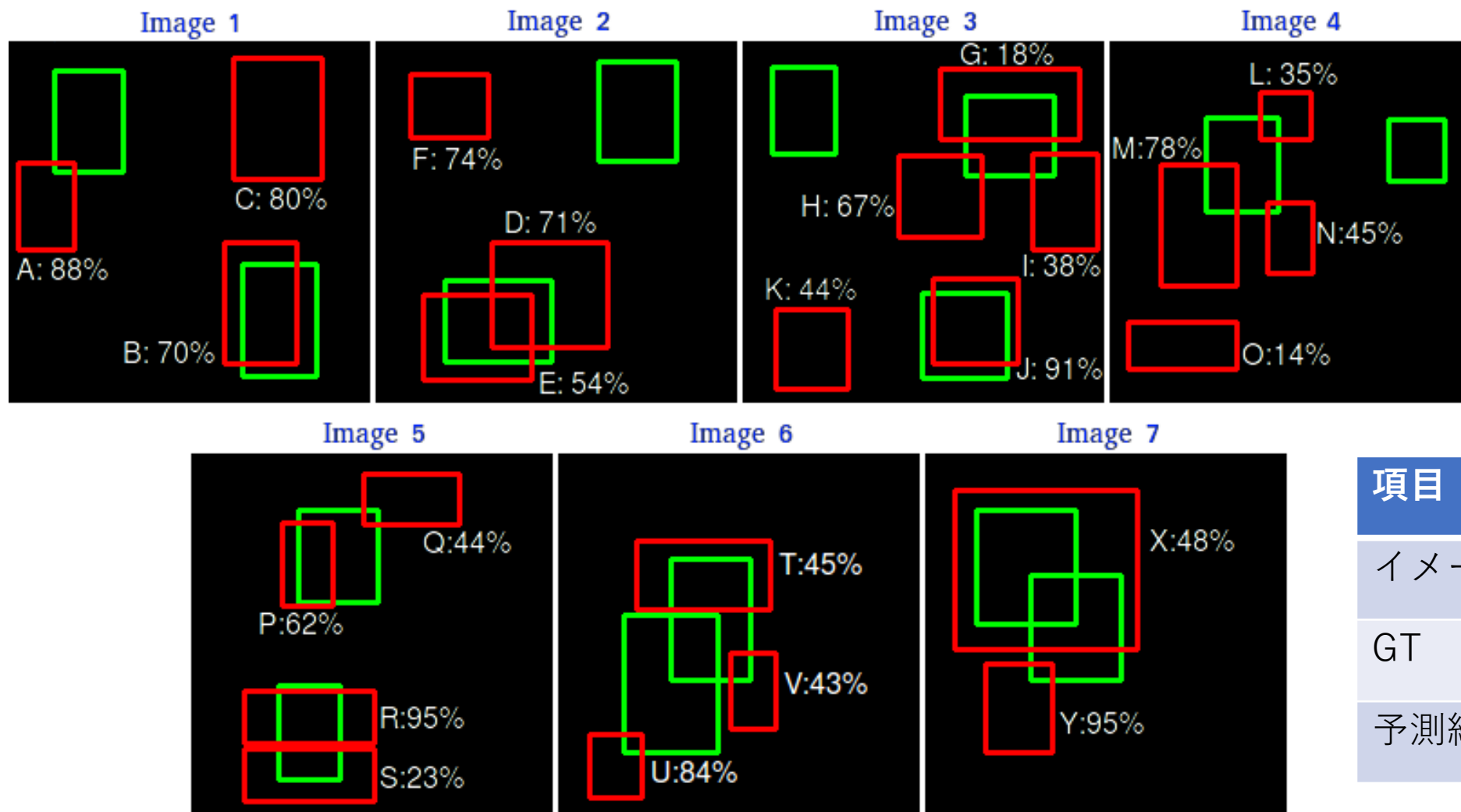
- 予測結果とGround Truthがどれだけ重なっているかを表す。



$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$



AP (Average Precision)



項目	合計
イメージ数	7
GT	15
予測結果	24

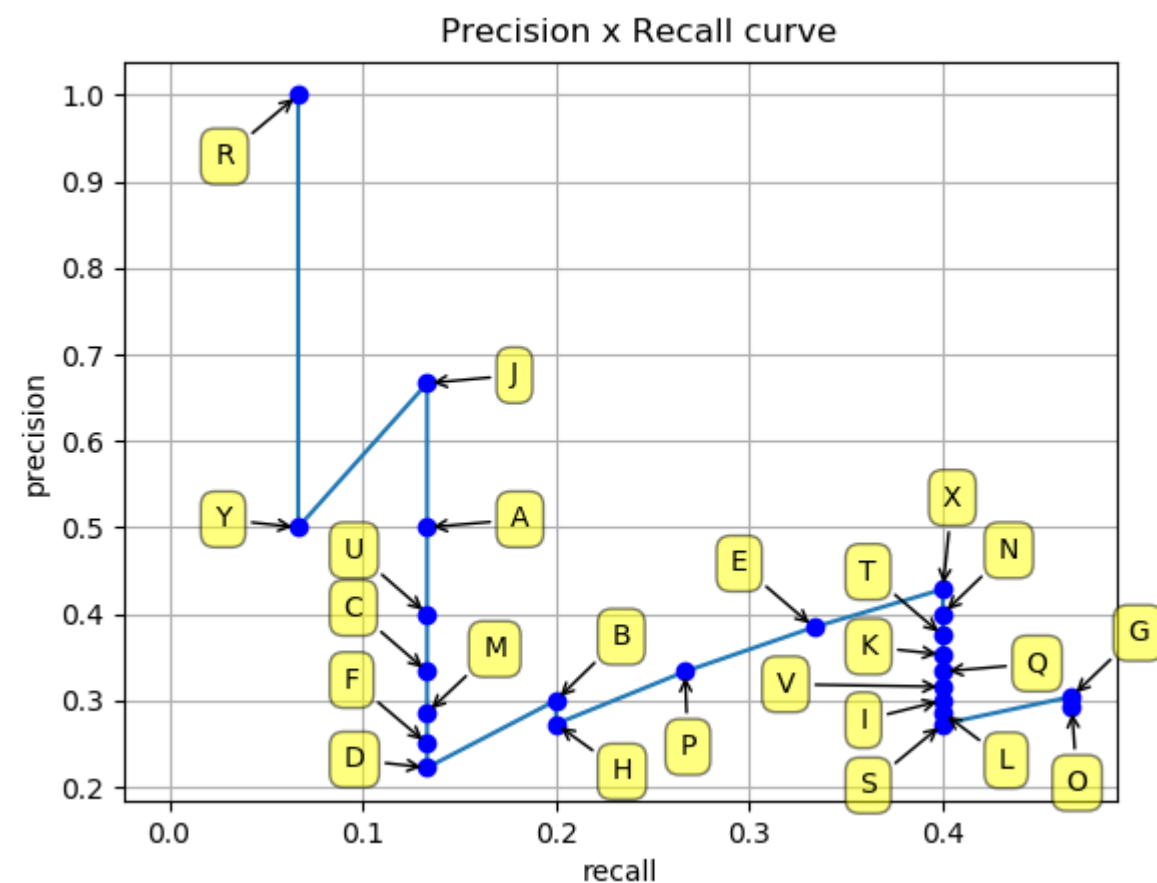
- TP : IoU > 30%
- FP : その他

この予測結果は
同じ種類

イメージ番号	予測結果	確信度	TP/FP	イメージ番号	予測結果	確信度	TP/FP
image1	A	88%	FP	image4	M	78%	FP
Image1	B	70%	TP	Image4	N	45%	FP
image1	C	80%	FP	Image4	O	14%	FP
image2	D	71%	FP	Image5	P	62%	TP
image2	E	54%	TP	Image5	Q	44%	FP
image2	F	74%	FP	Image5	R	95%	TP
image3	G	18%	TP	Image5	S	23%	FP
image3	H	67%	FP	Image6	T	45%	FP
image3	I	38%	FP	Image6	U	84%	FP
image3	J	91%	TP	Image6	V	43%	FP
image3	K	44%	FP	Image7	X	48%	TP
Image4	L	35%	FP	Image7	Y	95%	FP

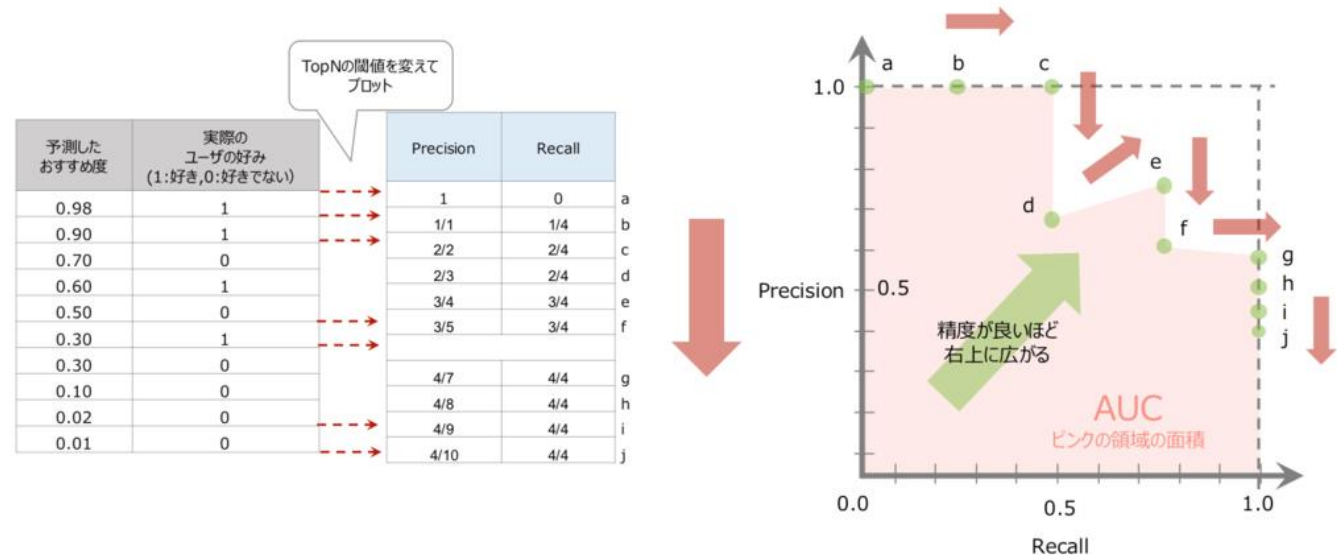
Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

確認度のソート順



PR曲線

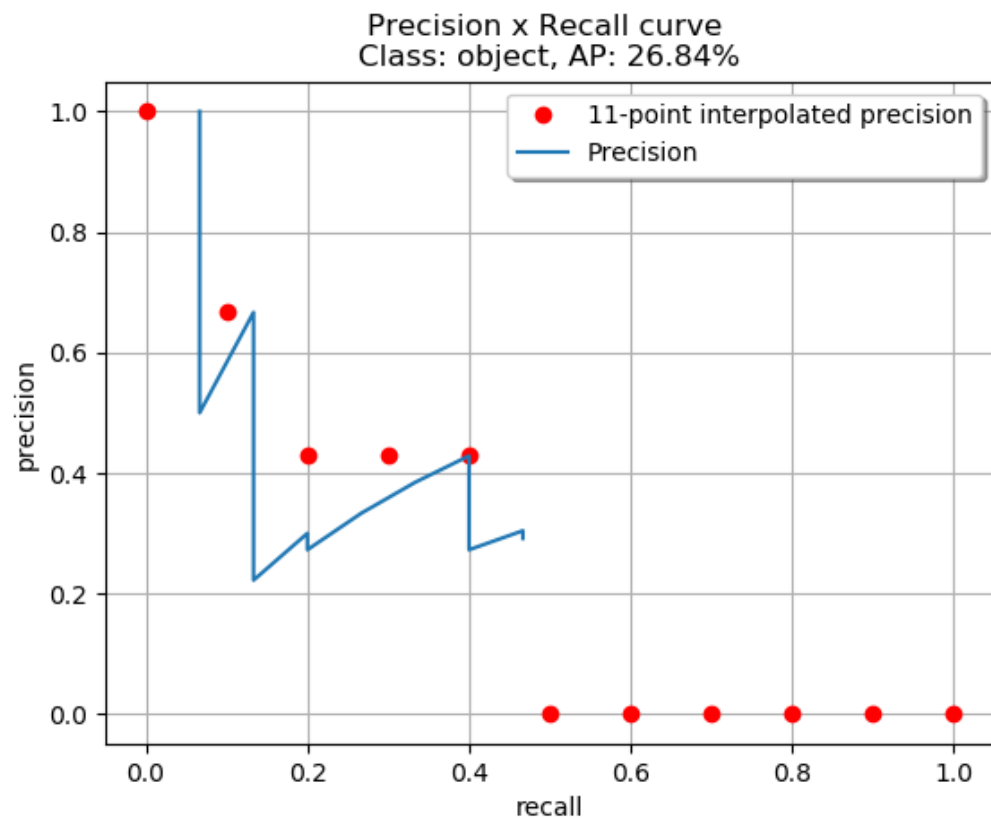
- Precision-Recall Curve
- Recallを横軸に、Precisionを縦軸にとり、Top1、Top2, … というように閾値であるTopNの Nを変動させると、RecallとPrecisionが複数点プロットできます。この点同士を直線で結んだものがPR曲線です。



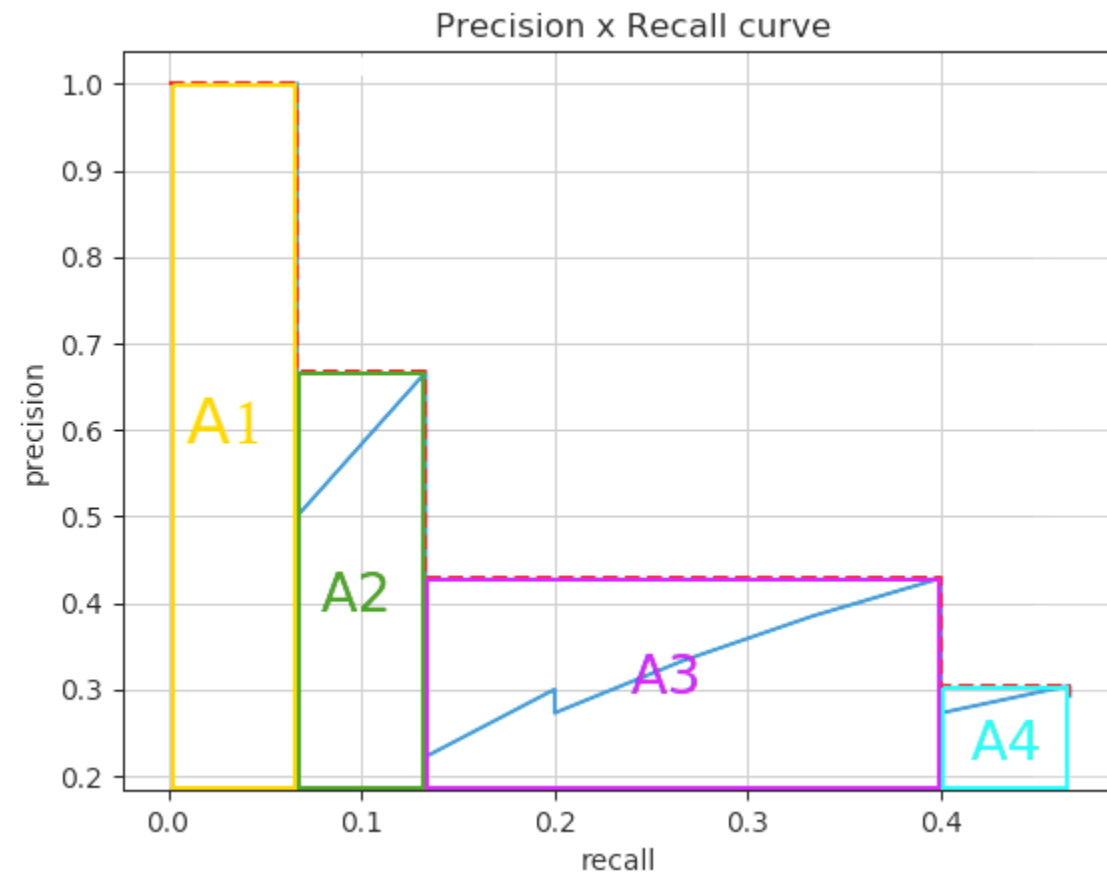
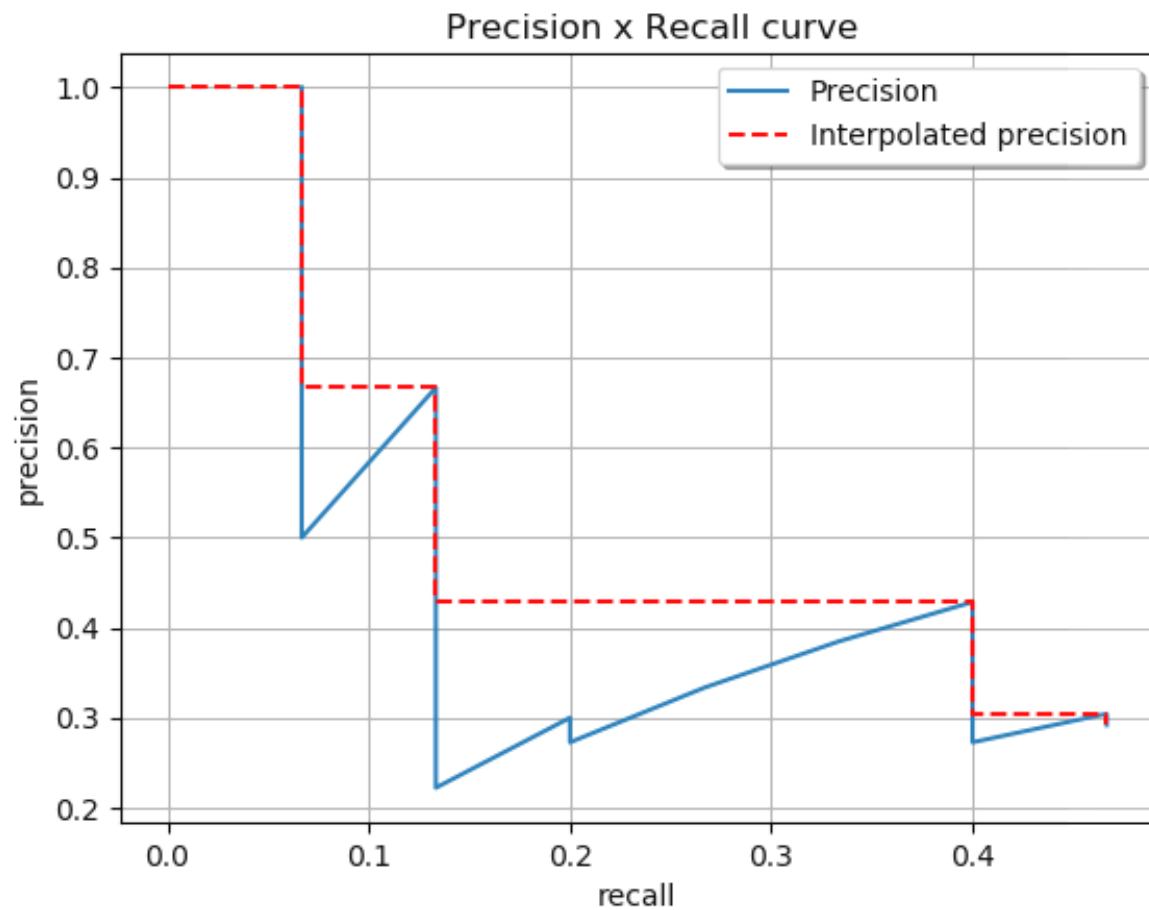
AP

- AP(Average Precision)は物体検出の精度を比較するための指標
- APは適合アイテムが出現した時点をそれぞれ閾値として、閾値ごとのPrecisionを算出し、Precisionの平均をとったものです。

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666



APの計算①



APの計算②

$$AP = A_1 + A_2 + A_3 + A_4$$

$$A_1 = (0.0666 - 0) * 1 = 0.0666$$

$$A_2 = (0.1333 - 0.0666) * 0.0666 = 0.04446222$$

$$A_3 = (0.4 - 0.1333) * 0.4285 = 0.11428095$$

$$A_4 = (0.4666 - 0.4) * 0.3043 = 0.02026638$$

$$AP = 0.0666 + 0.04446222 + 0.11428095 + 0.02026638$$

$$AP = 0.24560955 = 24.56\%$$

mAP

- 複数種類の場合、mAPを利用
- mAPは全てのクラスラベルのAPの平均である

SSD論文の記載

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
Fast [6]	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster [2]	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
Faster [2]	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
Faster [2]	07+12+COCO	78.8	84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9
SSD300	07	68.0	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8	69.2	76.6	82.1	77.0	72.5	41.2	64.2	69.1	78.0	68.5
SSD300	07+12	74.3	75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0
SSD300	07+12+COCO	79.6	80.9	86.3	79.0	76.2	57.6	87.3	88.2	88.6	60.5	85.4	76.7	87.5	89.2	84.5	81.4	55.0	81.9	81.5	85.9	78.9
SSD512	07	71.6	75.1	81.4	69.8	60.8	46.3	82.6	84.7	84.1	48.5	75.0	67.4	82.3	83.9	79.4	76.6	44.9	69.9	69.1	78.1	71.8
SSD512	07+12	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
SSD512	07+12+COCO	81.6	86.6	88.3	82.4	76.0	66.3	88.6	88.9	89.1	65.1	88.4	73.6	86.5	88.9	85.3	84.6	59.1	85.0	80.4	87.4	81.2

デプロイ

- アプリ作成
- SDK提供
- サービス提供

APIサービス例

- Flask
- Flask-Restful