CSCI 467: Machine Learning

Discussion – Cross-Validation and Evaluation Metrics

Slides by: Wang (Bill) Zhu (Sep 2023)
Modified by: Ameya Godbole (Feb 2024)
Materials from: *https://developers.google.com/machine-learning/crash-course*
Materials from: *https://scikit-learn.org/stable/modules/cross_validation.html*

Cross-Validation Overview

- Training and Test Sets
- Validation Set
- Cross-Validation

Training and Test Sets

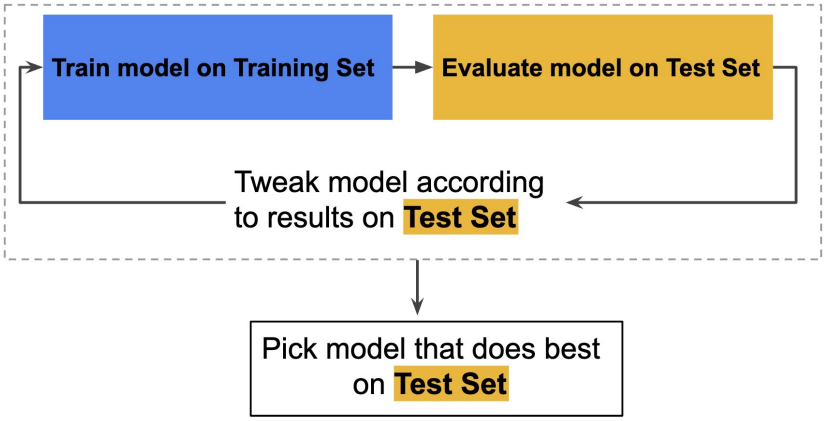
- Training set - a subset to train a model.
- Test set – a subset to test a trained model

You could imagine slicing the single data set as follows (80%/20%):

Image from: *https://developers.google.com/machine-learning/crash-course*

Training and Test Sets

- With two partitions, the workflow could look as follows (may overfit the test set)



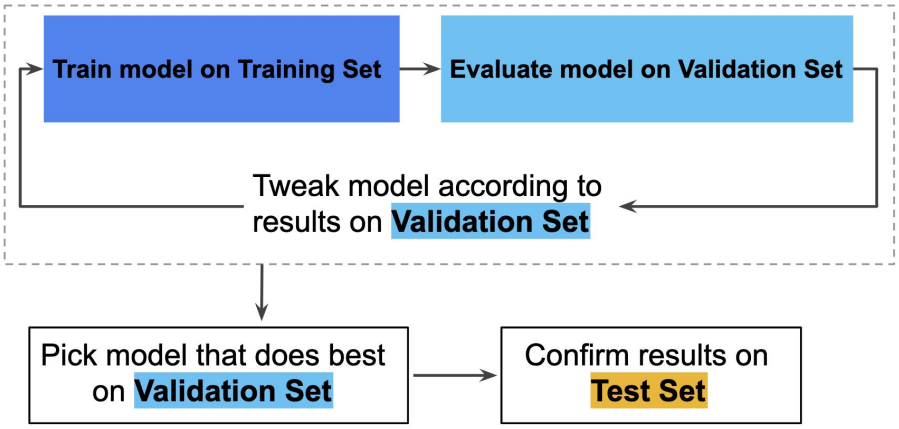
Validation Set

- You can greatly reduce your chances of overfitting by partitioning the data set into the three subsets shown in the following figure.
- Use the **validation set to evaluate results** from the training set. Then, use the **test set to double-check your evaluation** after the model has "passed" the validation set. (exam analogy: Lectures, HWs, Finals)

Image from: *https://developers.google.com/machine-learning/crash-course*

Validation Set

- Tune hyper-parameters (batch size, learning rate, etc.) on the validation set



Cross-Validation

- You need the validation set to be large (avoid overfitting)
- You need the validation set to be small (to have enough training data)

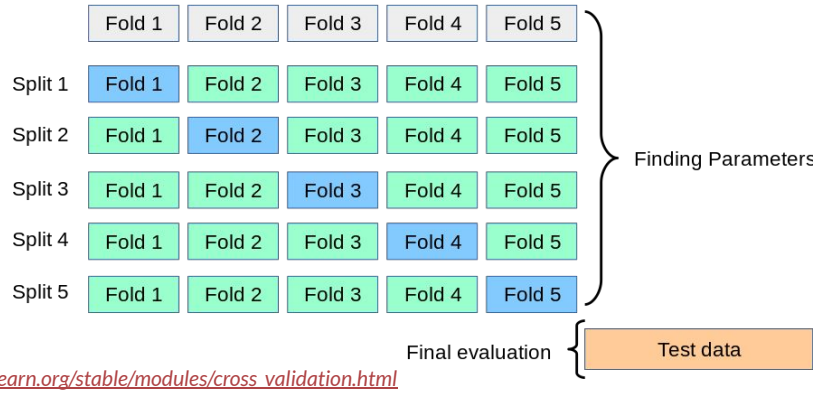


Image from: https://scikit-learn.org/stable/modules/cross_validation.html

Cross-Validation

- Split the data into k fold, use (k-1) fold for training and 1 fold for validation
- After finalizing hyper-parameters, use the entire training+validation to train the model

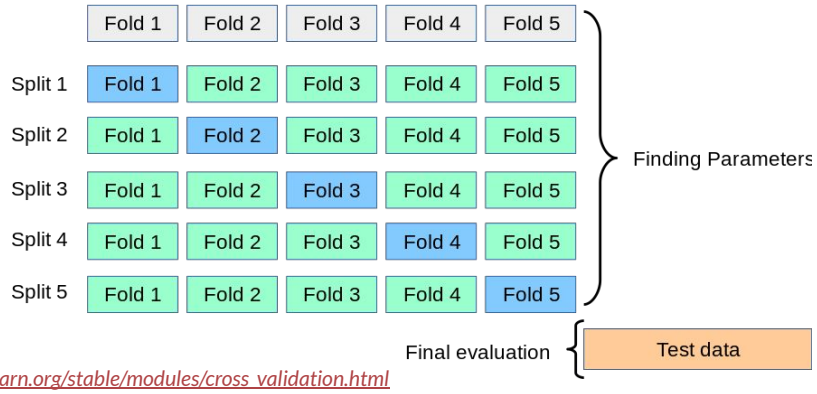


Image from: https://scikit-learn.org/stable/modules/cross_validation.html

More on this topic

How do we sample validation sets for imbalance classes?

> Stratified K-fold Cross-Validation

- *https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html#sklearn.model_selection.StratifiedKFold*

Evaluation Metrics Overview

Classifying Examples

- Thresholding
- Confusion matrix

Evaluation

- Accuracy
- Precision and Recall
- ROC and AUC
- Calibration

Thresholding

- Binary classification: $y = f(x), y \in \{0, 1\}$
- A logistic regression model outputs a probability in $(0, 1)$

- Logistic regression returns a probability. You have to convert the returned probability to a binary value (for example, this email is spam).
- A logistic regression model that returns 99.9% for a particular email message is predicting that it is very likely to be spam. Conversely, another email message with a prediction score of 0.03% on that same logistic regression model is very likely not spam. However, what about an email message with a prediction score of 0.6? In order to map a logistic regression value to a binary category, you must define a **classification threshold** (also called the **decision threshold**). A value above that threshold indicates "spam"; a value below indicates "not spam." It is tempting to assume that the classification threshold should always be 0.5, but thresholds are problem-dependent, and are therefore values that you must tune.
- The following sections take a closer look at metrics you can use to evaluate a classification model's predictions, as well as the impact of changing the classification threshold on these predictions.

Thresholding

- Binary classification: $y = f(x), y \in \{0, 1\}$
- A logistic regression model outputs a probability in $(0, 1)$
- Choose a threshold to convert it to a binary value
- 0.5 is not always the best
- *Why? Depends on the evaluation metrics.*

- Logistic regression returns a probability. You have to convert the returned probability to a binary value (for example, this email is spam).
- A logistic regression model that returns 99.9% for a particular email message is predicting that it is very likely to be spam. Conversely, another email message with a prediction score of 0.03% on that same logistic regression model is very likely not spam. However, what about an email message with a prediction score of 0.6? In order to map a logistic regression value to a binary category, you must define a **classification threshold** (also called the **decision threshold**). A value above that threshold indicates "spam"; a value below indicates "not spam." It is tempting to assume that the classification threshold should always be 0.5, but thresholds are problem-dependent, and are therefore values that you must tune.
- The following sections take a closer look at metrics you can use to evaluate a classification model's predictions, as well as the impact of changing the classification threshold on these predictions.

Confusion Matrix – Tumor Prediction

- Use 2x2 confusion matrix to separate out different kinds of errors
- Class-imbalanced setup: 9% of examined tumors are malignant, 91% benign

True Positives (TP) Reality: Malignant ML predicted: Malignant	**False Positives (FP)** Reality: Benign ML predicted: Malignant Type-1 Error
False Negatives (FN) Reality: Malignant ML predicted: Benign Type-2 Error	**True Negatives (TN)** Reality: Benign ML predicted: Benign

- We can summarize the tumor prediction model using a 2x2 confusion matrix that depicts all four possible outcomes
- True vs. False – your prediction is true or false
- Positive vs. Negative – your prediction is positive or negative
- Connection to some stats courses: FP and FN also called Type-1 and Type-2 errors
- In the following sections, we'll look at how to evaluate classification models using metrics derived from these four outcomes.

Evaluation Metrics: Accuracy - Can Be Misleading

- Accuracy is the fraction of predictions our model got right

True Positives (TP) Reality: Malignant ML predicted: Malignant Number of TP results: 1	**False Positives (FP)** Reality: Benign ML predicted: Malignant Number of FP results: 1
False Negatives (FN) Reality: Malignant ML predicted: Benign Number of FN results: 8	**True Negatives (TN)** Reality: Benign ML predicted: Benign Number of TN results: 90

- Of the 100 tumor examples, 91 are benign (90 TNs and 1 FP) and 9 are malignant (1 TP and 8 FNs).
- Of the 91 benign tumors, the model correctly identifies 90 as benign. That's good. However, of the 9 malignant tumors, the model only correctly identifies 1 as malignant—a terrible outcome, as 8 out of 9 malignancies go undiagnosed!
- While 91% accuracy may seem good at first glance, another tumor-classifier model that always predicts benign would achieve the exact same accuracy (91/100 correct predictions) on our examples. In other words, our model is no better than one that has zero predictive ability to distinguish malignant tumors from benign tumors.
- Accuracy alone doesn't tell the full story when you're working with a **class-imbalanced data set**, like this one, where there is a significant disparity between the number of positive and negative labels.
- In the next section, we'll look at two better metrics for evaluating class-imbalanced problems: precision and recall

Evaluation Metrics: Accuracy - Can Be Misleading

- Accuracy is the fraction of predictions our model got right
- Accuracy $= \frac{TP+TN}{TP+FP+FN+TN}$

True Positives (TP) Reality: Malignant ML predicted: Malignant Number of TP results: 1	**False Positives (FP)** Reality: Benign ML predicted: Malignant Number of FP results: 1
False Negatives (FN) Reality: Malignant ML predicted: Benign Number of FN results: 8	**True Negatives (TN)** Reality: Benign ML predicted: Benign Number of TN results: 90

- Of the 100 tumor examples, 91 are benign (90 TNs and 1 FP) and 9 are malignant (1 TP and 8 FNs).
- Of the 91 benign tumors, the model correctly identifies 90 as benign. That's good. However, of the 9 malignant tumors, the model only correctly identifies 1 as malignant—a terrible outcome, as 8 out of 9 malignancies go undiagnosed!
- While 91% accuracy may seem good at first glance, another tumor-classifier model that always predicts benign would achieve the exact same accuracy (91/100 correct predictions) on our examples. In other words, our model is no better than one that has zero predictive ability to distinguish malignant tumors from benign tumors.
- Accuracy alone doesn't tell the full story when you're working with a **class-imbalanced data set**, like this one, where there is a significant disparity between the number of positive and negative labels.
- In the next section, we'll look at two better metrics for evaluating class-imbalanced problems: precision and recall

Evaluation Metrics: Accuracy - Can Be Misleading

- Accuracy is the fraction of predictions our model got right
- Accuracy $= \frac{TP+TN}{TP+FP+FN+TN}$
- How about a model that predicts negative all the time?

True Positives (TP) Reality: Malignant ML predicted: Malignant Number of TP results: 1	**False Positives (FP)** Reality: Benign ML predicted: Malignant Number of FP results: 1
False Negatives (FN) Reality: Malignant ML predicted: Benign Number of FN results: 8	**True Negatives (TN)** Reality: Benign ML predicted: Benign Number of TN results: 90

- Of the 100 tumor examples, 91 are benign (90 TNs and 1 FP) and 9 are malignant (1 TP and 8 FNs).
- Of the 91 benign tumors, the model correctly identifies 90 as benign. That's good. However, of the 9 malignant tumors, the model only correctly identifies 1 as malignant—a terrible outcome, as 8 out of 9 malignancies go undiagnosed!
- While 91% accuracy may seem good at first glance, another tumor-classifier model that always predicts benign would achieve the exact same accuracy (91/100 correct predictions) on our examples. In other words, our model is no better than one that has zero predictive ability to distinguish malignant tumors from benign tumors.
- Accuracy alone doesn't tell the full story when you're working with a **class-imbalanced data set**, like this one, where there is a significant disparity between the number of positive and negative labels.
- In the next section, we'll look at two better metrics for evaluating class-imbalanced problems: precision and recall

Exercise (2 mins)

In which of the following scenarios would suggest that the ML model is doing a good job?

A. A deadly, but curable, medical condition afflicts .01% of the population. An ML model uses symptoms as features and predicts this affliction with an accuracy of 99.99%.

B. An expensive robotic chicken crosses a very busy road a thousand times per day. An ML model evaluates traffic patterns and predicts when this chicken can safely cross the street with an accuracy of 99.99%.

C. In the game of roulette, a ball is dropped on a spinning wheel and eventually lands in one of 38 slots. Using visual features (the spin of the ball, the position of the wheel when the ball was dropped, the height of the ball over the wheel), an ML model can predict the slot that the ball will land in with an accuracy of 50%.

. C

Evaluation Metrics: Precision and Recall

- What proportion of positive identifications was actually correct?
- Precision $= \frac{TP}{TP+FP}$

True Positives (TP) Reality: Malignant ML predicted: Malignant Number of TP results: 1	**False Positives (FP)** Reality: Benign ML predicted: Malignant Number of FP results: 1
False Negatives (FN) Reality: Malignant ML predicted: Benign Number of FN results: 8	**True Negatives (TN)** Reality: Benign ML predicted: Benign Number of TN results: 90

- Precision: What proportion of positive identifications was actually correct?
- Recall: What proportion of actual positives was identified correctly?

Evaluation Metrics: Precision and Recall

- What proportion of positive identifications was actually correct?
- Precision $= \frac{TP}{TP+FP}$
- 0.5

True Positives (TP) Reality: Malignant ML predicted: Malignant Number of TP results: 1	**False Positives (FP)** Reality: Benign ML predicted: Malignant Number of FP results: 1
False Negatives (FN) Reality: Malignant ML predicted: Benign Number of FN results: 8	**True Negatives (TN)** Reality: Benign ML predicted: Benign Number of TN results: 90

- Precision: What proportion of positive identifications was actually correct?
- Recall: What proportion of actual positives was identified correctly?

Evaluation Metrics: Precision and Recall

- What proportion of actual positives was identified correctly?
- Recall $= \frac{TP}{TP+FN}$

True Positives (TP) Reality: Malignant ML predicted: Malignant Number of TP results: 1	**False Positives (FP)** Reality: Benign ML predicted: Malignant Number of FP results: 1
False Negatives (FN) Reality: Malignant ML predicted: Benign Number of FN results: 8	**True Negatives (TN)** Reality: Benign ML predicted: Benign Number of TN results: 90

- Precision: What proportion of positive identifications was actually correct?
- Recall: What proportion of actual positives was identified correctly?

Evaluation Metrics: Precision and Recall

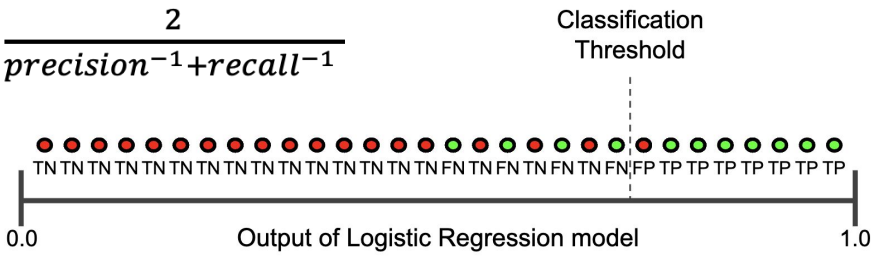
- What proportion of actual positives was identified correctly?
- Recall $= \frac{TP}{TP+FN}$
- 0.11

True Positives (TP) Reality: Malignant ML predicted: Malignant Number of TP results: 1	**False Positives (FP)** Reality: Benign ML predicted: Malignant Number of FP results: 1
False Negatives (FN) Reality: Malignant ML predicted: Benign Number of FN results: 8	**True Negatives (TN)** Reality: Benign ML predicted: Benign Number of TN results: 90

- Precision: What proportion of positive identifications was actually correct?
- Recall: What proportion of actual positives was identified correctly?

Precision and Recall: A Tug of War

- Hard to optimize both at the same time by changing threshold
- Precision $= \frac{TP}{TP+FP}$, Recall $= \frac{TP}{TP+FN}$
- F1 $= \frac{2}{precision^{-1}+recall^{-1}}$



- Unfortunately, precision and recall are often in tension. That is, improving precision typically reduces recall and vice versa. Explore this notion by looking at the following figure, which shows 30 predictions made by an email classification model. Those to the right of the classification threshold are classified as "spam", while those to the left are classified as "not spam.

Exercise (2 min)

Consider a classification model that separates email into two categories: "spam" or "not spam." If you raise the classification threshold, what will happen to precision?

A. Probably increase.
B. Probably decrease.
C. Definitely increase.
D. Definitely decrease.

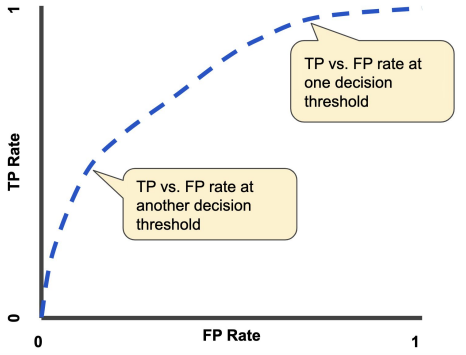
Consider two models—A and B—that each evaluate the same dataset. Which one of the following statements is true?

A. If model A has better recall than model B, then model A is better.
B. If model A has better precision and better recall than model B, then model A is probably better.
C. If Model A has better precision than model B, then model A is better.

- A, B (imagine moving the threshold on the previous page)
- B

A ROC Curve

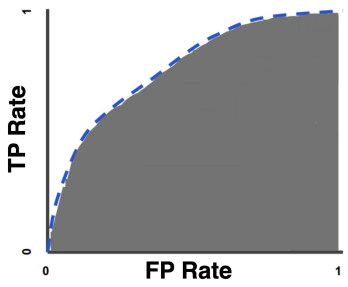
- Each point is the TP and FP rate at one decision threshold
- TPR (Recall) $= \frac{TP}{TP+FN}$
- FPR $= \frac{FP}{FP+TN}$

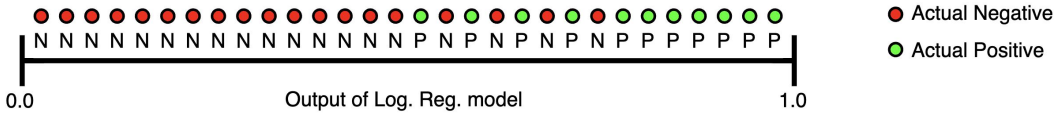


- **True Positive Rate** (**TPR**) is a synonym for recall
- **False Positive Rate** (**FPR**) is defined
- Receiver operating characteristic curve
- An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.

Evaluation Metrics: AUC (AUROC)

- AUC: “Area under the ROC Curve”
- The probability that the model ranks a random positive example more highly than a random negative example
- Independent of the threshold

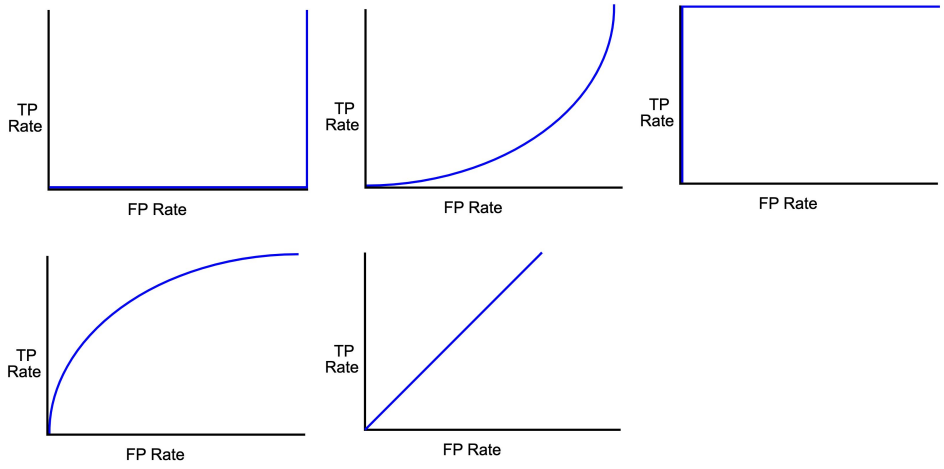




- AUC is **scale-invariant**. It measures how well predictions are ranked, rather than their absolute values.
- AUC is **classification-threshold-invariant**. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.
- Intuition: gives an aggregate measure of performance aggregated across all possible classification thresholds

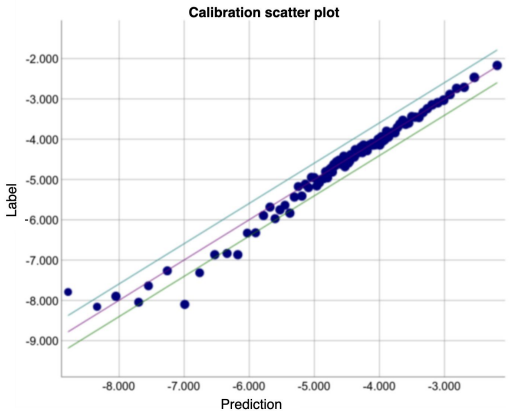
Exercise (2 mins)

Which of the following ROC curves produce AUC values greater than 0.5?



Calibration

- Prediction bias = average of prediction - average of labels
- Zero bias alone does not mean everything is perfect
- It's a great sanity check: incomplete features? noisy data? buggy pipeline?
- Don't fix bias with a calibration layer, fix it in the model



- A significant nonzero prediction bias tells you there is a bug somewhere in your model, as it indicates that the model is wrong about how frequently positive labels occur.
- For example, let's say we know that on average, 1% of all emails are spam. If we don't know anything at all about a given email, we should predict that it's 1% likely to be spam. Similarly, a good spam model should predict on average that emails are 1% likely to be spam. (In other words, if we average the predicted likelihoods of each individual email being spam, the result should be 1%.) If instead, the model's average prediction is 20% likelihood of being spam, we can conclude that it exhibits prediction bias.
- Possible root causes of prediction bias are:
  - Incomplete feature set
  - Noisy data set
  - Buggy pipeline
  - Biased training sample
  - Overly strong regularization

Tuning Hyperparameters

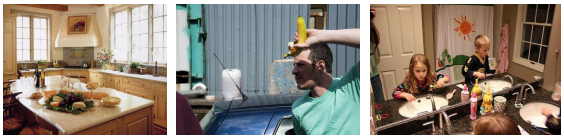
- Grid Search
  - Define the of values a hyperparam can take
  - Iteratively try each value
- Random Search
  - Define the of values a hyperparam can take
  - For continuous values, define a range
  - Randomly sample a value for each hyperparam
- Bayesian Hyperparam Optimization
  - [Beyond the scope of this discussion]

Is this it for evaluation?

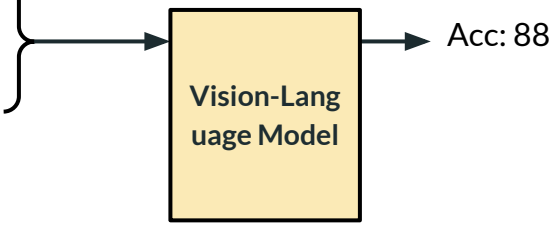
We haven't looked at what is in the data…

Application: Contrast set

*Q: Is there **at least** 1 image with exactly 2 dark bottles on a counter.*

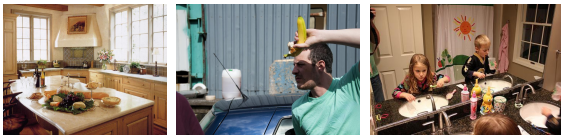


Expected A: True

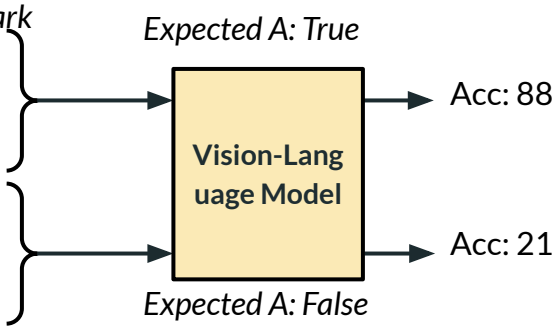


Application: Contrast set

*Q: Is there **at least** 1 image with exactly 2 dark bottles on a counter.*



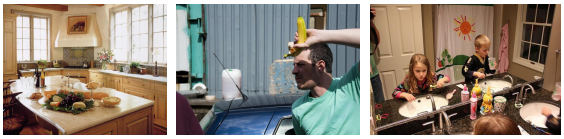
*Contrast Q: Is there **less than** 1 image with exactly 2 dark bottles on a counter.*



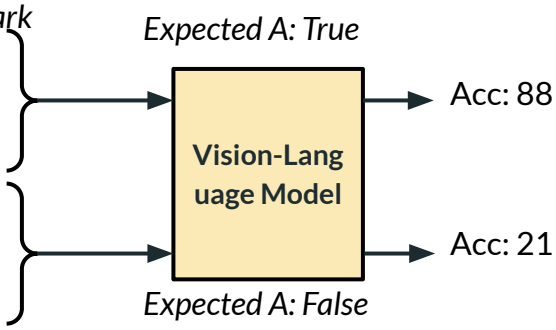
Example from: https://arxiv.org/abs/2210.15037

Application: Contrast set

*Q: Is there **at least** 1 image with exactly 2 dark bottles on a counter.*



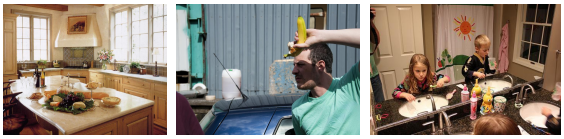
*Contrast Q: Is there **less than** 1 image with exactly 2 dark bottles on a counter.*



What does this tell us? Contrast Qs are hard? They have low correlation/grounding on images? The VL model is bad?

Application: Contrast set

*Q: Is there **at least** 1 image with exactly 2 dark bottles on a counter.*



*Contrast Q: Is there **less than** 1 image with exactly 2 dark bottles on a counter.*

Expected A: True

Vision-Language Model

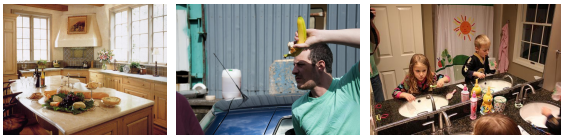
Expected A: False

TP: 80	**FP:** 11
FN: 25	**TN:** 188

TP: 20	**FP:** 83
FN: 157	**TN:** 44

Application: Contrast set

*Q: Is there **at least** 1 image with exactly 2 dark bottles on a counter.*



*Contrast Q: Is there **less than** 1 image with exactly 2 dark bottles on a counter.*

Expected A: True

Vision-Language Model

Expected A: False

TP: 80	**FP:** 11
FN: 25	**TN:** 188

TP: 20	**FP:** 83
FN: 157	**TN:** 44

What does this tell us? (Probably) the model is over-stable on its prediction.

Example from: https://arxiv.org/abs/2210.15037