Foundation of Machine Learning

Day 2: Evaluation and Improvement for Machine Learning Algorithms

Sieu Tran

Agenda

Day 2 of the Foundation of Machine Learning course will focus on evaluation and improvement of ML models

1. Recap if Day 1
2. Introduction to ML Evaluation
3. Data Train/Test/Validation Procedure
4. Bias/Variance Trade-off and Regularization
5. Evaluating and Improving Regression Models
6. Evaluating and Improving Classification Models
7. Other Evaluation Metrics
8. Closing Words

Recap

An Introduction Machine Learning Evaluation

Debugging A Learning Algorithm

Key facts (situation): Suppose you have implemented regularized linear regression to predict housing prices.

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{m}\theta_j^2\right]$$

Key question: However, when you test your hypothesis on a new set of houses, you find that it makes unacceptably large errors in its predictions. What should you try next?

Solutions:

- Get more training examples
- Try smaller sets of features
- Try getting additional features
- Try adding polynomial features x_1^2, x_2^2, x_1x_2, etc.
- Try decreasing λ
- Try increasing λ

Machine Learning Diagnostic

DIAGNOSTIC

- A test that you can run to gain insight what is/isn't working with a learning algorithm, and
- Gain guidance as to how best to improve its performance.

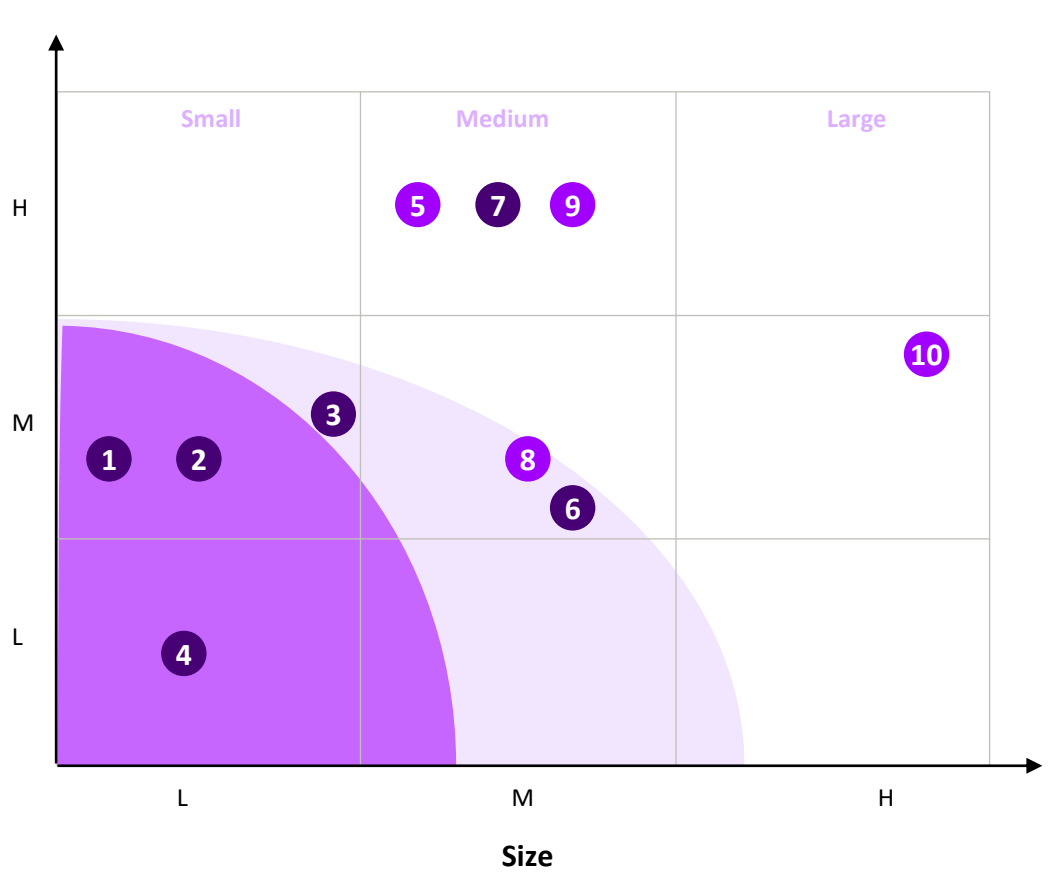
TIME INVESTMENT

- Diagnostics can take time to implement, but
- Doing so can be a very good use of your time.

Evaluating Your Hypothesis

Fails to generalize to new examples not in training set.

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$



List of variables:

1. Size Of House
2. # Of Bedrooms
3. # Of Floors
4. Age Of House
5. Average Income In Neighborhood
6. Kitchen Size

⋮

100

Evaluating Your Hypothesis

Train

Size	Price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300

Test

1534	315
1427	199
1380	212
1494	243

$$\left(x^{(1)}, y^{(1)}\right)$$
$$\left(x^{(2)}, y^{(2)}\right)$$
$$\left(x^{(3)}, y^{(3)}\right)$$
$$\left(x^{(4)}, y^{(4)}\right)$$
$$\dots$$
$$\left(x^{(m)}, y^{(m)}\right)$$

$$\left(x_{\text{Test}}^{(1)}, y_{\text{Test}}^{(1)}\right)$$
$$\left(x_{\text{Test}}^{(2)}, y_{\text{Test}}^{(2)}\right)$$
$$\dots$$
$$\left(x_{\text{Test}}^{(n)}, y_{\text{Test}}^{(n)}\right)$$

Training/Testing Procedure For Logistic Regression

First step:

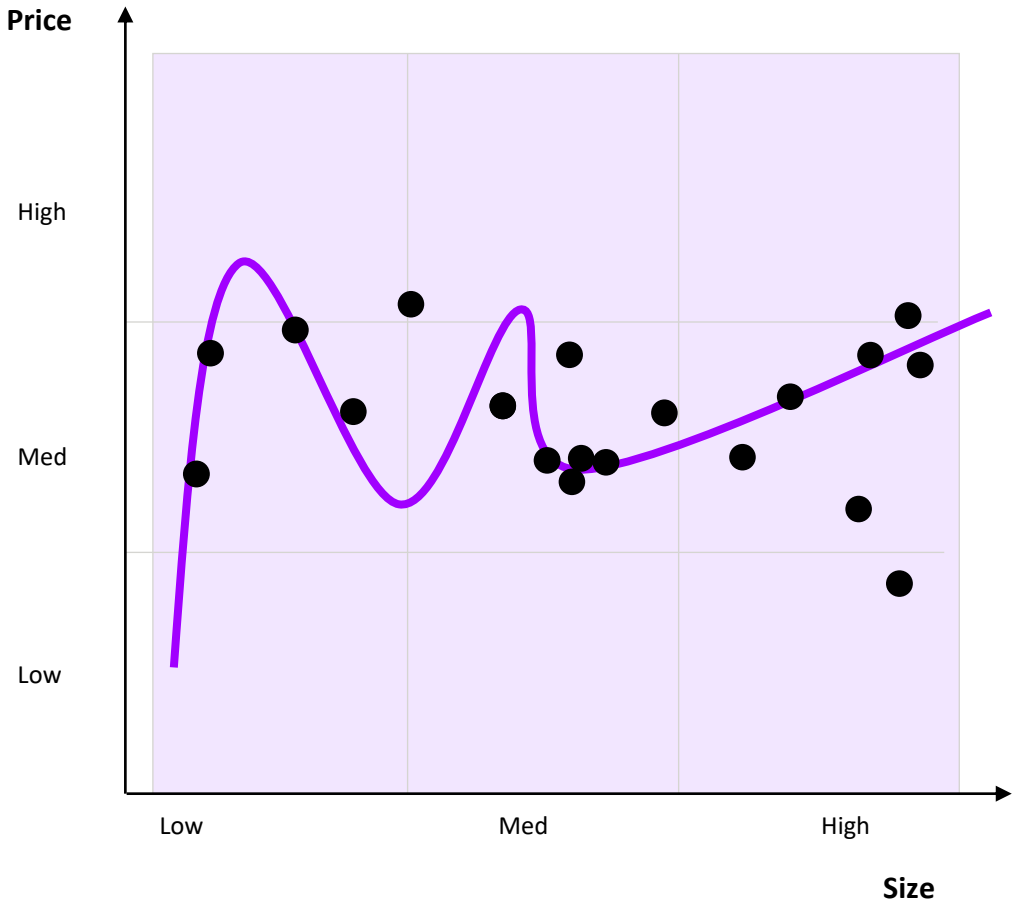
Learn parameter θ from training data

Second step:

Compute test set error

Overfitting Example

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$



Initial Attempt:

Once parameters $\theta_0, \ldots, \theta_4$ were fit to some set of data (training set), the error of the parameters as measured on that data (the training error $J(\theta)$) is likely to be lower than the actual generalization error.

Model Selection

Choose:

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5$$

Question:

How well does the model generalize?

Report test set error $J_{\text{Test}}(\theta^{(5)})$

Problem:

$J_{\text{Test}}(\theta^{(5)})$ is likely to be an optimistic estimate of generalization error, i.e., our extra parameter (degree of polynomial) is fit to test set.

Different Model Choices:

1. $h_\theta(x) = \theta_0 + \theta_1 x$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$
3. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$

⋮

10. $h_\theta(x) = \theta_0 + \theta_1 x + \ \ldots + \theta_{10} x^{10}$

Evaluating Your Hypothesis

Size	Price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	212
1494	243

$$(x^{(1)}, y^{(1)})$$
$$(x^{(2)}, y^{(2)})$$
$$\vdots$$
$$(x^{(m)}, y^{(m)})$$

$$(x_{cv}^{(1)}, y_{cv}^{(1)})$$
$$(x_{cv}^{(2)}, y_{cv}^{(2)})$$
$$\vdots$$
$$(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$$

$$(x_{test}^{(1)}, y_{test}^{(1)})$$
$$(x_{test}^{(2)}, y_{test}^{(2)})$$
$$\vdots$$
$$(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$$

Train/Validation/Test Error

Training error:

Formula

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Cross Validation error:

Formula

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

Test error:

Formula

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

Model Selection

Model Options

1. $h_\theta(x) = \theta_0 + \theta_1 x$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$
3. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$

⋮

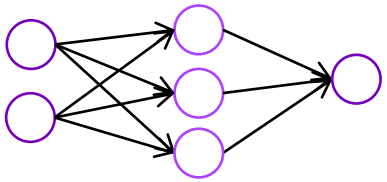
10. $h_\theta(x) = \theta_0 + \theta_1 x + \ldots + \theta_{10} x^{10}$

Select Model

- Pick $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_4 x^4$
- Estimate generalization error for test set $J_{\text{Test}}(\theta^{(5)})$

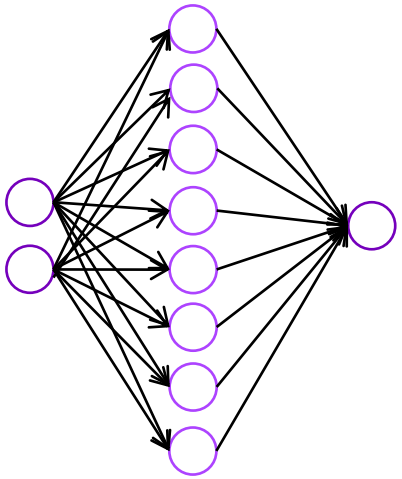
Neural Networks And Overfitting

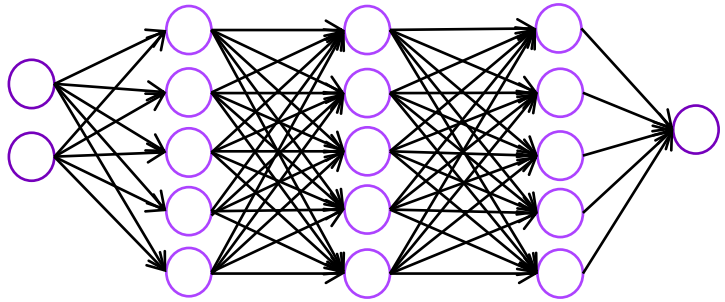
“Small” neural network
- **Fewer parameters**
- **More prone to underfitting**



Computationally cheaper

“Large” neural network
- **More parameters**
- **More prone to overfitting**





Computationally more expensive.
Use regularization (λ) to address overfitting.

Train/Valid/Test Data Procedure

Definitions

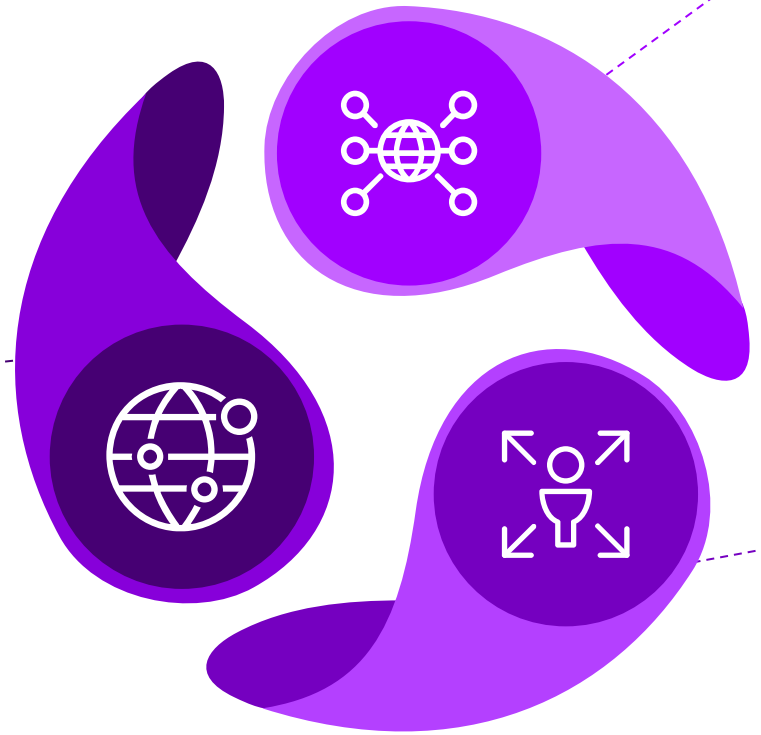
Train Dataset

→ Set of data used for learning (by the model), that is,

→ to fit the parameters to the machine learning model

Test Dataset

→ Set of data used to provide an unbiased evaluation of a final model fitted on the training dataset.



Validation Dataset

→ Set of data used to provide an unbiased evaluation of a model fitted on the training dataset while tuning model hyperparameters.

→ Also play a role in other forms of model preparation, such as feature selection, threshold cut-off selection.

The Science Behind Dataset Split Ratio

Question:

- Often it is asked in what proportion to split your dataset into Train, Validation, and Test sets?

This decision mainly depends on two things:

- The total number of samples in your data, and
- On the actual model you are training.

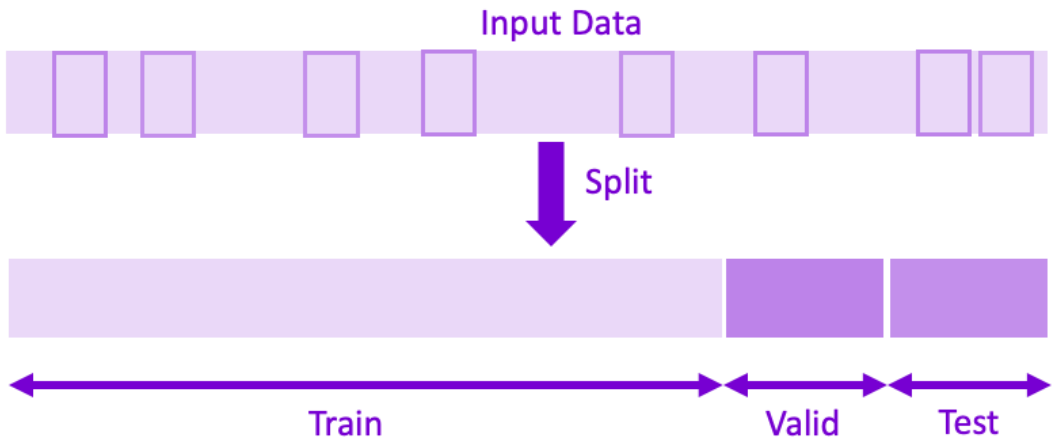
Potential Problems:

- Some models need substantial data to train upon, so you would optimize for the more extensive training sets in this case.
- Yan probably reduce the size of your validation set.
- If your model has many hyper-parameters, you would want to have a significant validation set as well.
- You probably don't need a validation set.

Approach

- Splitting randomly
- Splitting using the temporal component

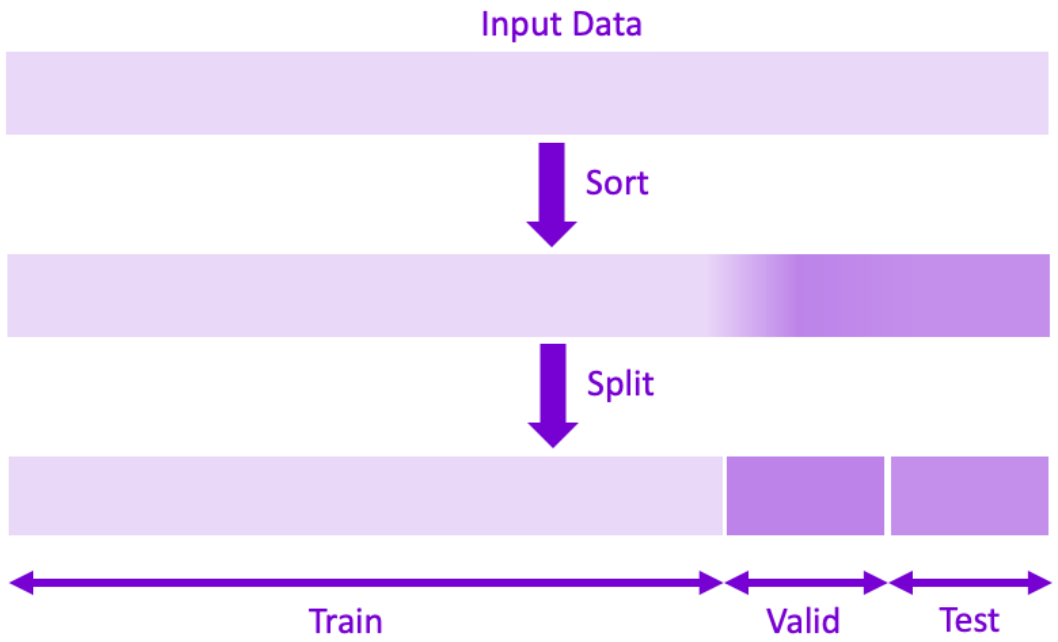
Splitting Randomly



Strategy:

1. You can't evaluate the predictive performance of a model with the same data you used for training.
2. It would be best if you evaluated the model with **new data that hasn't been seen** by the model before.
3. **Randomly splitting** the data is the most commonly used method for that **unbiased evaluation**.

Splitting Using The Temporal Component

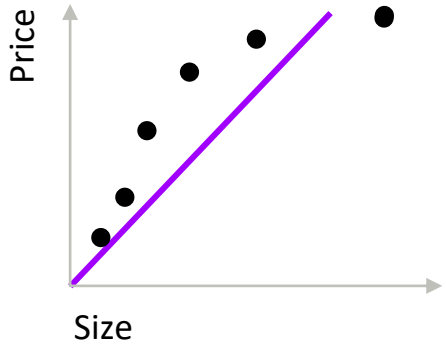


Strategy:

1. Using that temporal variable is a more reliable way of splitting datasets whenever the dataset includes the date variable, and the objective is to predict something in the future.
2. Hence, use the latest samples for creating the validation and test dataset.
3. The main idea is always choosing a subset of samples representing the data faithfully in our model will receive afterward.

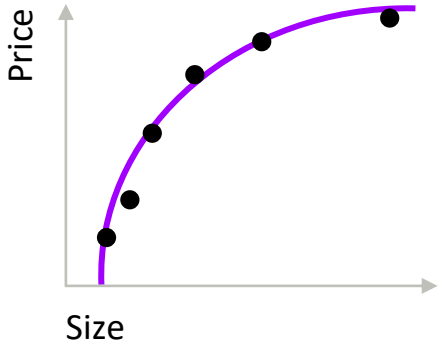
Addressing Bias/Variance via Regularization

Bias/Variance



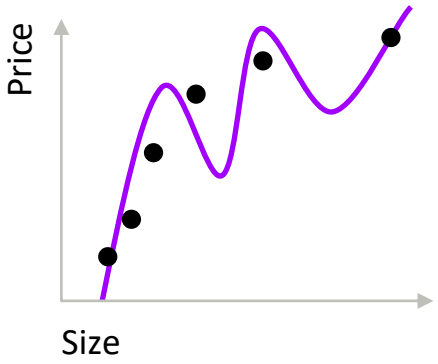
$\theta_0 + \theta_1 x$

High Bias (Underfit)



$\theta_0 + \theta_1 x + \theta_2 x^2$

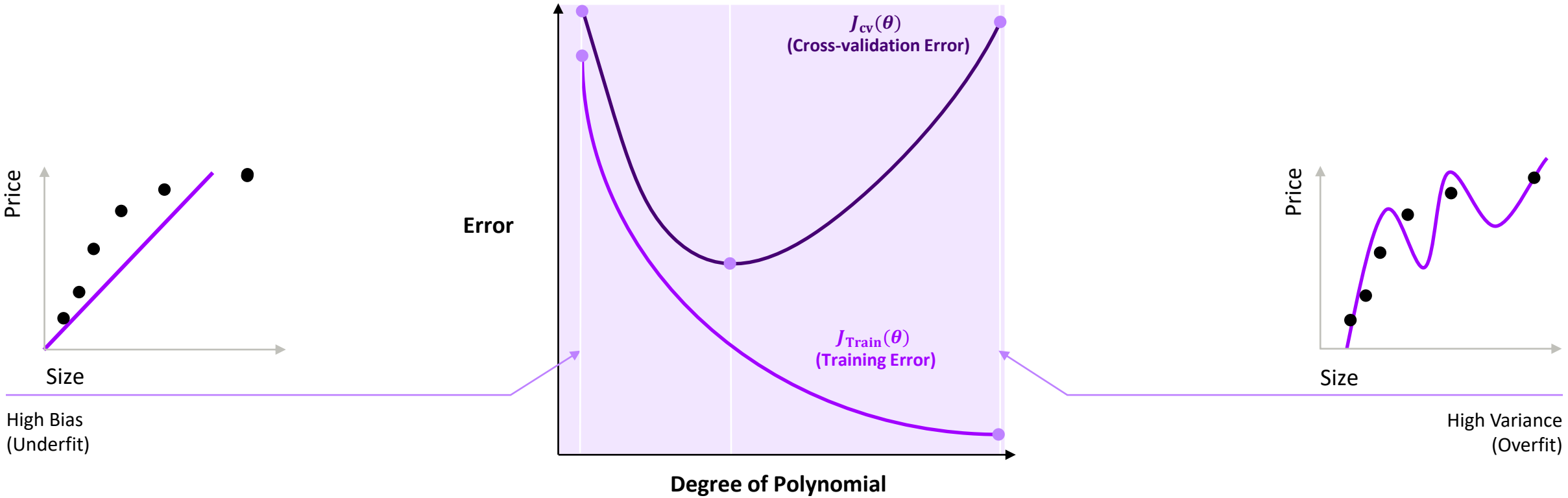
“Just Right”



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High Variance (Overfit)

Bias/Variance



Training error: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

Cross validation error: $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$

Diagnosing Bias vs. Variance

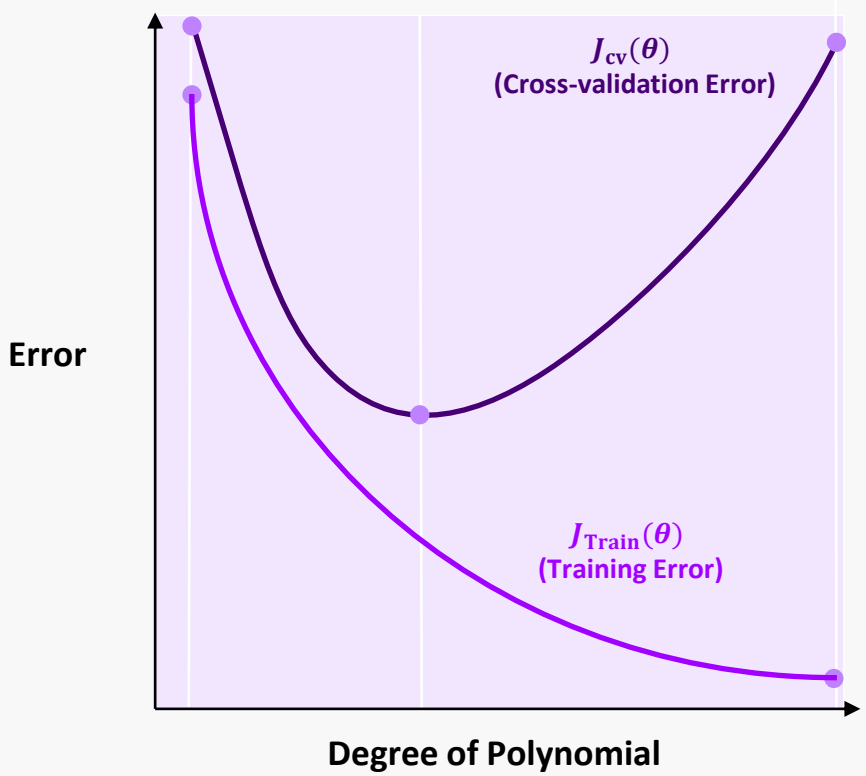
Question:

- Suppose your learning algorithm is performing less well than you were hoping ($J_{\text{cv}}(\theta)$ or $J_{\text{Test}}(\theta)$ is high). Is it a bias problem or a variance problem?

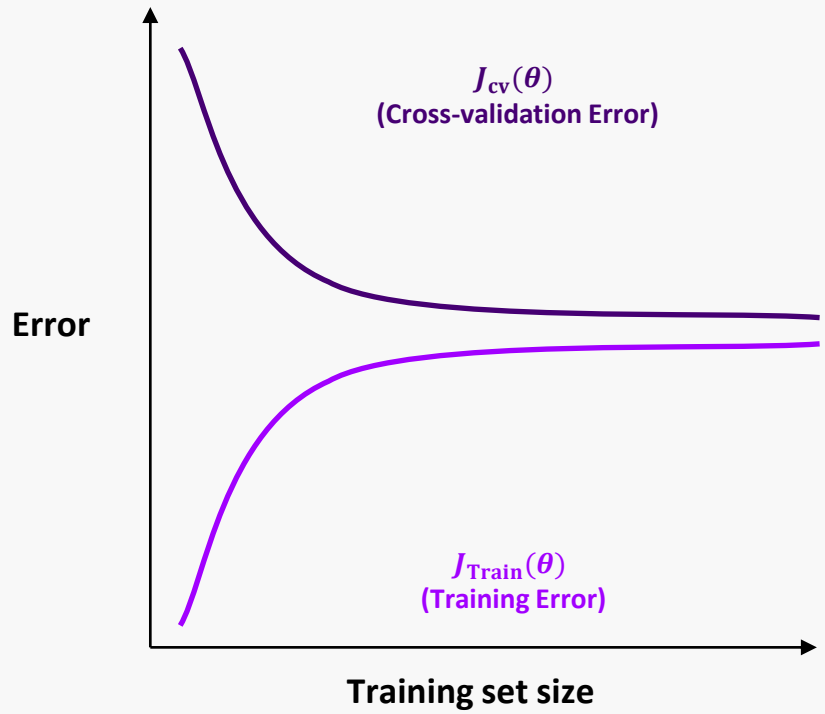
How to diagnose:

- High bias (underfit):
  - $J_{\text{Train}}(\theta)$ will be high and
  - $J_{\text{Train}}(\theta) \approx J_{\text{cv}}(\theta)$
- High variance (overfit):
  - $J_{\text{Train}}(\theta)$ will be low and
  - $J_{\text{Train}}(\theta) << J_{\text{cv}}(\theta)$

Error vs. Degree Plot

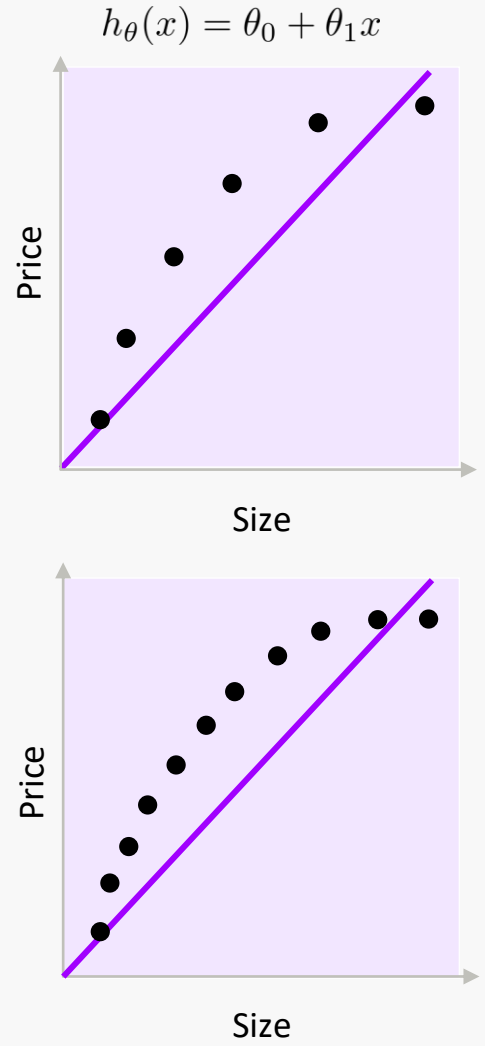


High Bias

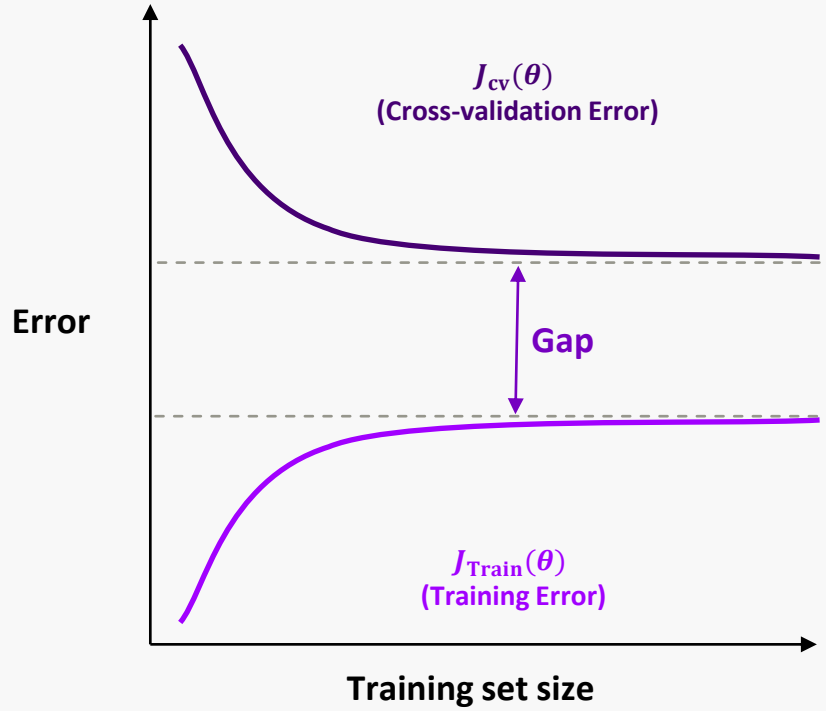


Disclaimer:

If a learning algorithm is suffering from high bias, **getting more training data will not (by itself) help much**.



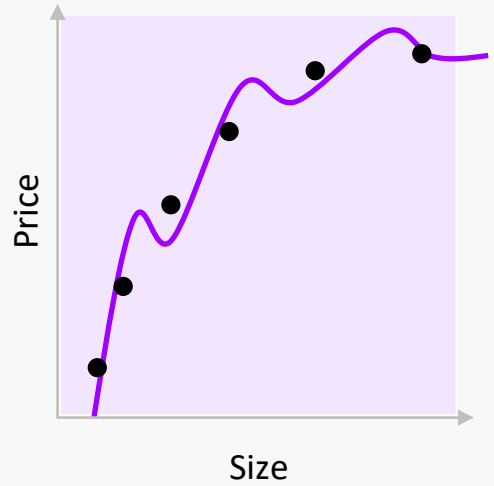
High Variance

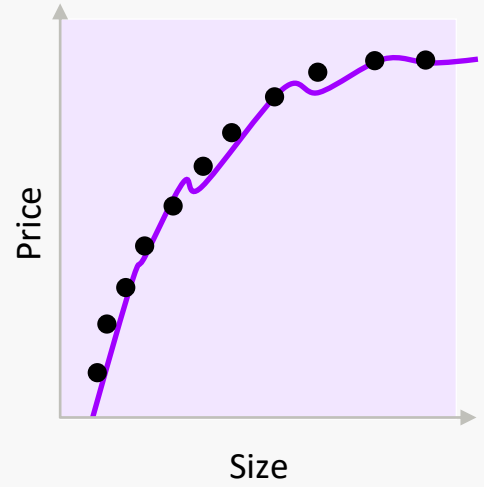


Disclaimer:

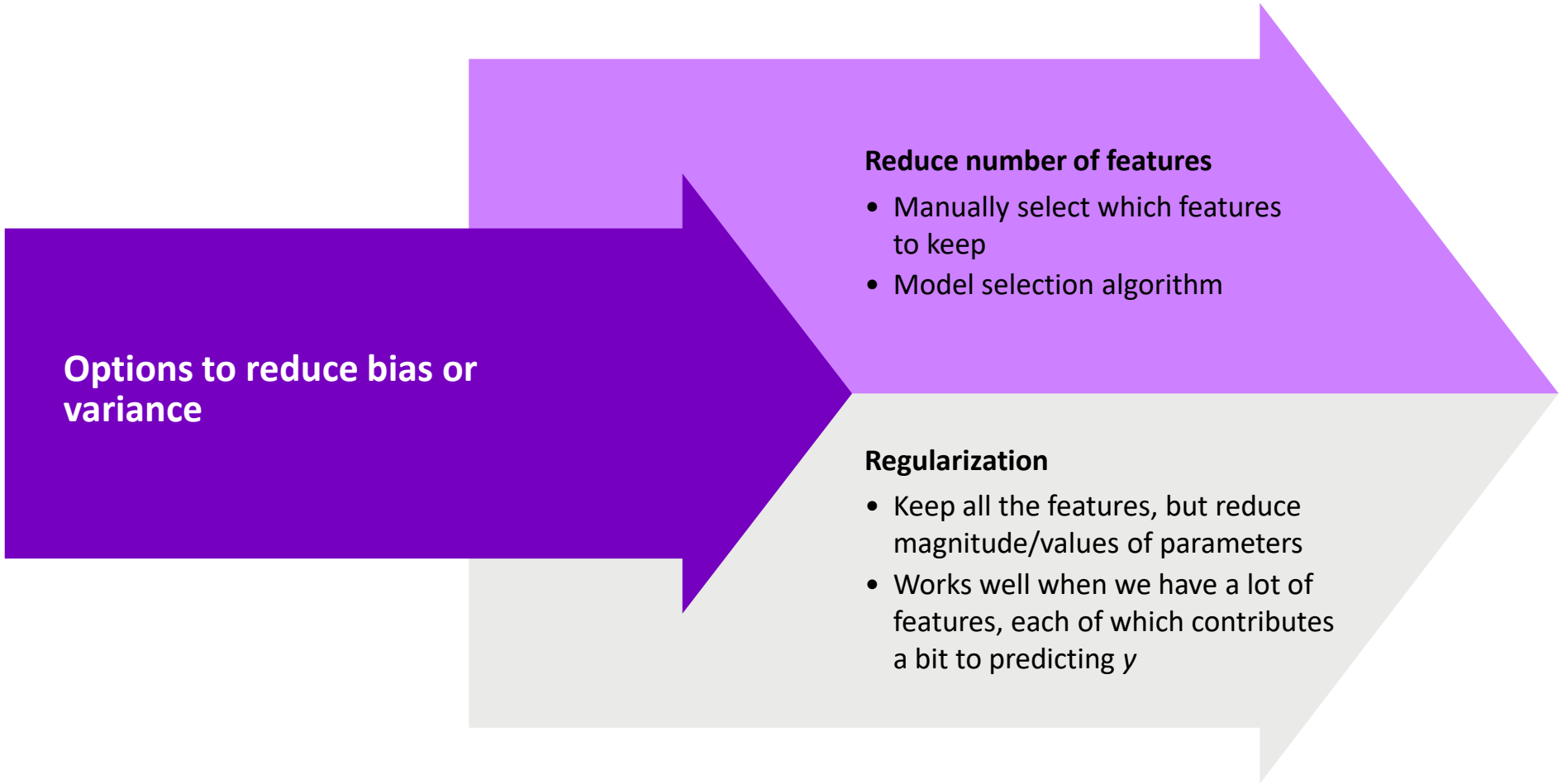
If a learning algorithm is suffering from high variance, **getting more training data is likely to help**.

$$h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{100}$$





Addressing Bias and Variance Problem



What Is Regularization?

Regularization

Small values for parameters $\theta_0, \ldots, \theta_n$	Housing example
• "Simpler" hypothesis • Less prone to overfitting	• Features: $x_1, \ldots, x_{100}$ • Parameters: $\theta_0, \ldots, \theta_{100}$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]$$

Implementation Plan For Regularization

Consideration for each phase

Phase 1

- In regularization we use the θ to minimize

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n}\theta_j^2\right]$$

Review

Phase 2

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?

- Algorithm works fine; setting λ to be very large can't hurt it
- Algorithm fails to eliminate overfitting.
- Algorithm results in underfitting. (Fails to fit even training data well).
- Gradient descent will fail to converge.

Review

Phase 3

Recalibrate or find a different approach

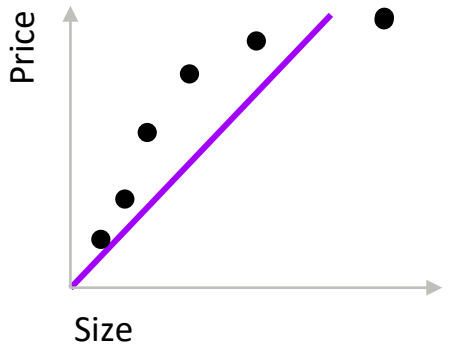
- Choosing the regularization parameter λ
- Compute the test error $J(\theta)$
- Evaluate

Review

Linear Regression With Regularization

Model: $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

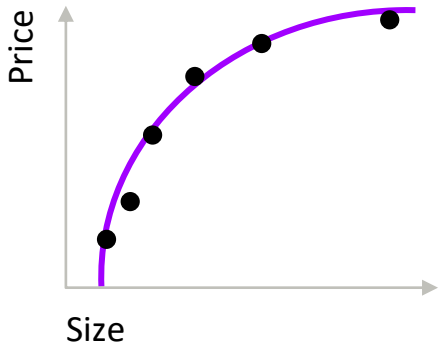


$\theta_0 + \theta_1 x$

$\lambda = 10000.\ \theta_1 \approx 0, \theta_2 \approx 0, \ldots$

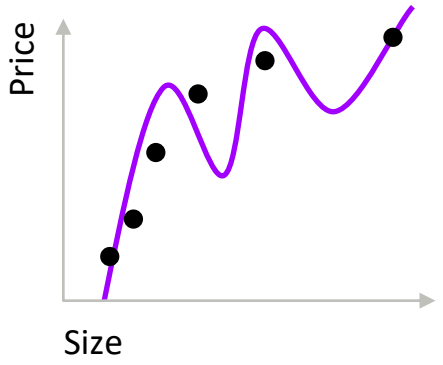
$h_\theta(x) \approx \theta_0$

Large λ
High Bias
(Underfit)



$\theta_0 + \theta_1 x + \theta_2 x^2$

Intermediate λ
"Just Right"

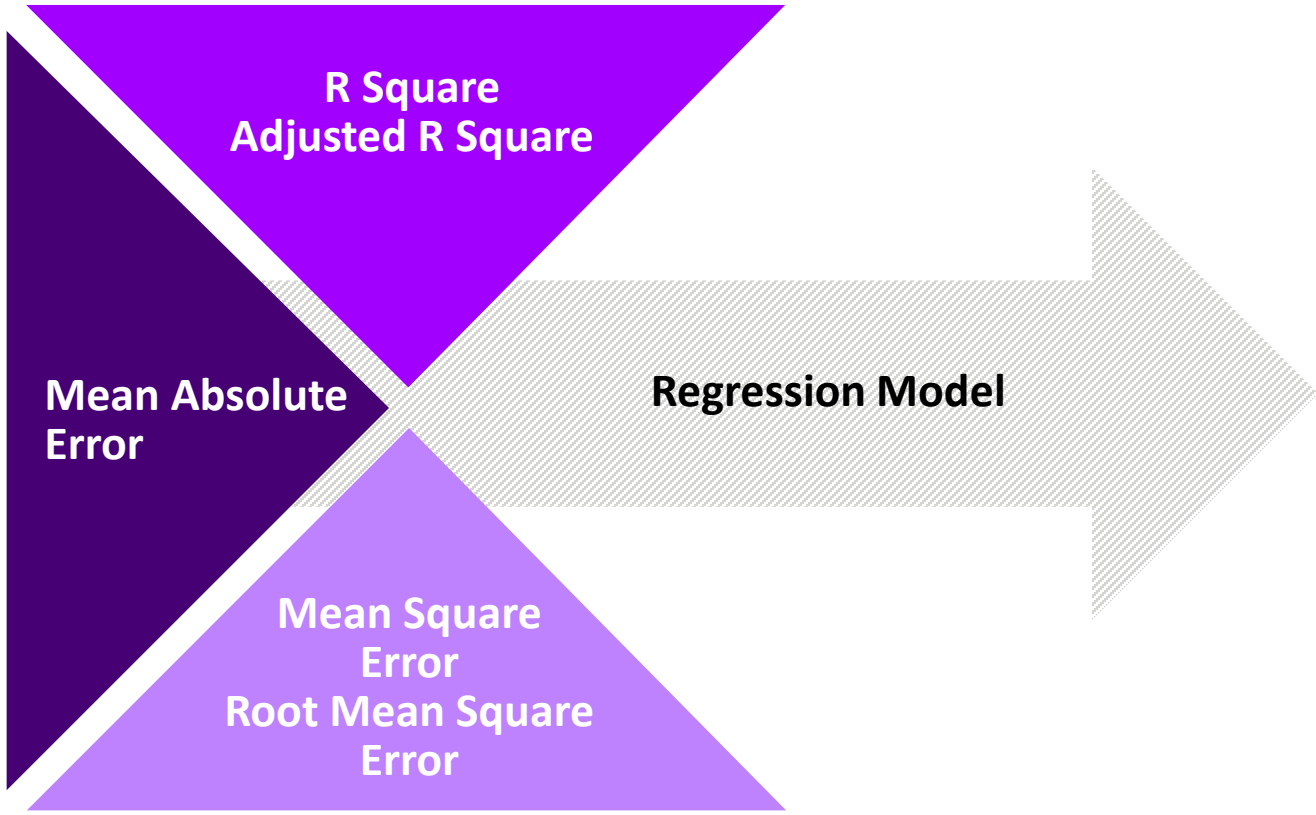


$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Small λ
High Variance
(Overfit)

Evaluating and Improving Regression Models

Regression Model Evaluation Metric



R Square & Adjusted R Square

Intuition

- R Square value is between 0 to 1 and a bigger value indicates a better fit between prediction and actual value.
- It is the square of the Correlation Coefficient.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Definitions

- R Square measures how much variability in dependent variable can be explained by the model.
- R Square is a good measure to determine how well the model fits the dependent variables.
- Adjusted R Square will penalize additional independent variables added to the model and adjust the metric to prevent overfitting issues.

R^2 & Adjusted R^2

Advantages

- Can to compare models more easily than MSE and RMSE.
- Adjusted R^2 has a degree over R^2 score since the adjusted R^2 score penalizes the addition of too many variable in the model.
- Can help indicate if a model is overfitting with a low adjusted R^2 score.

Disadvantages

- Models which overfit data are not penalized and would have a very high R^2 score.
- Models with bias can still have a relatively high R^2 score.

Mean Square Error (MSE) & Root Mean Square Error(RMSE)

Key facts:

- An absolute measure of the goodness for the fit. It gives you an absolute number on how much your predicted results deviate from the actual number.
- Root Mean Square Error (RMSE) is the square root of MSE.

  It is used more commonly than MSE because firstly sometimes MSE value can be too big to compare easily.

  Secondly, MSE is calculated by the square of error, and thus square root brings it back to the same level of prediction error and makes it easier for interpretation.

Formula:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Advantages

The mean squared error is a differentiable function and can be used as a loss function which can be minimized.

RMSE shares the same unit as y, which can be easier to interpret than MSE.

Disadvantages

The error can be quite difficult to interpret, what value of MSE is considered as acceptable?

Effected by outliers

As with MSE, RMSE is affected by outliers which may show worse model performance than if the model was fitted on data excluding outliers.

Mean Absolute Error (MAE)

Formula

$$\mathbf{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Advantage

- Do not square the errors; thus, not effected heavily by outliers.

Mean Absolute Error
(MAE)

Disadvantage

- The MAE is not always differentiable; thus, in some cases, can not be used as a loss function in regression models.

Definitions

- Instead of the sum of square of error in MSE, MAE is taking the sum of the absolute value of error.
- Compare to MSE or RMSE, MAE is a more direct representation of sum of error terms. MSE gives larger penalization to big prediction error by square it while MAE treats all errors the same.

The Akaike Information Criterion (AIC)

Definition

How to use

This is another quality of fit metric that takes into account the ability of the model to fit the data.

Provides a method for assessing the quality of a model through comparison of related models.

It's based on the Deviance metric, but penalizes you for making the model more complicated.

Intuition

It's useful for comparing models but isn't interpretable on its own.

The lower the AIC score the better.

✓ Much like adjusted R-squared, its intent is to prevent you from including irrelevant predictors.

✗ Unlike adjusted R-squared, the number itself is not meaningful. If you have more than one similar candidate models, then you should select the model that has the smallest AIC.

Residual Plots From Linear Regression Models

Patterns Emerge

- They're symmetrically distributed, tending to cluster towards the middle of the plot.
- They're clustered around the lower single digits of the *y*-axis.

Ambiguous

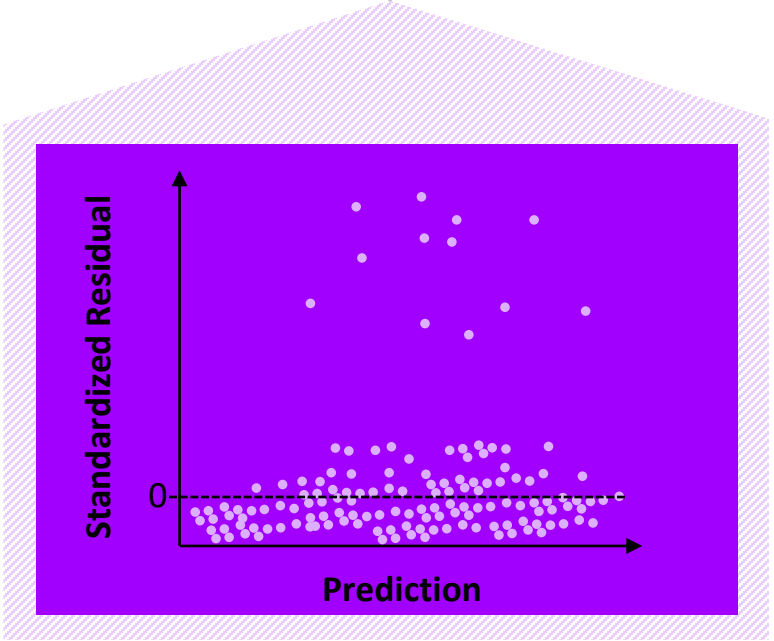
- In general, there aren't any clear patterns.

Positive, negative, or **zero value** for the residual respectively mean the prediction was **too low**, **too high**, or **exactly correct**.

Residual = Observed – Predicted

Y-axis Unbalanced

Example



Solution

POTENTIAL SOLUTIONS

- The solution to this is almost always to transform your data.
- It's also possible that your model lacks a variable.

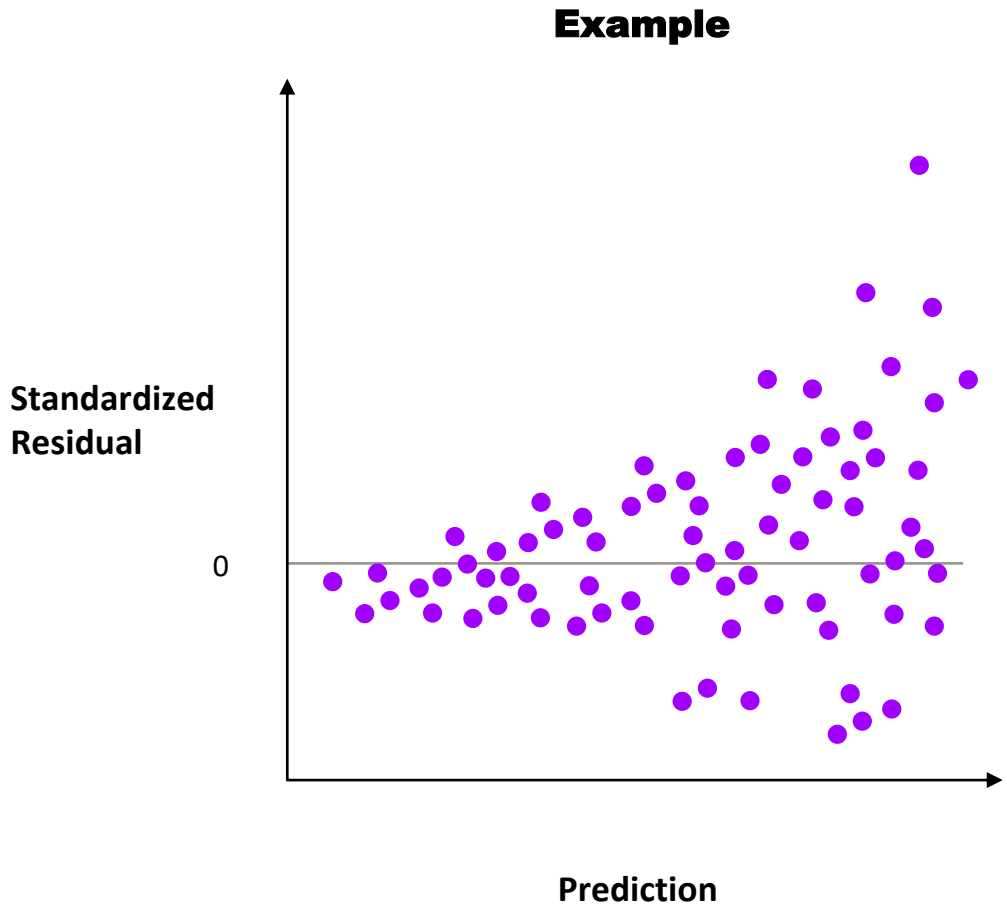
Heteroscedasticity

Implication:

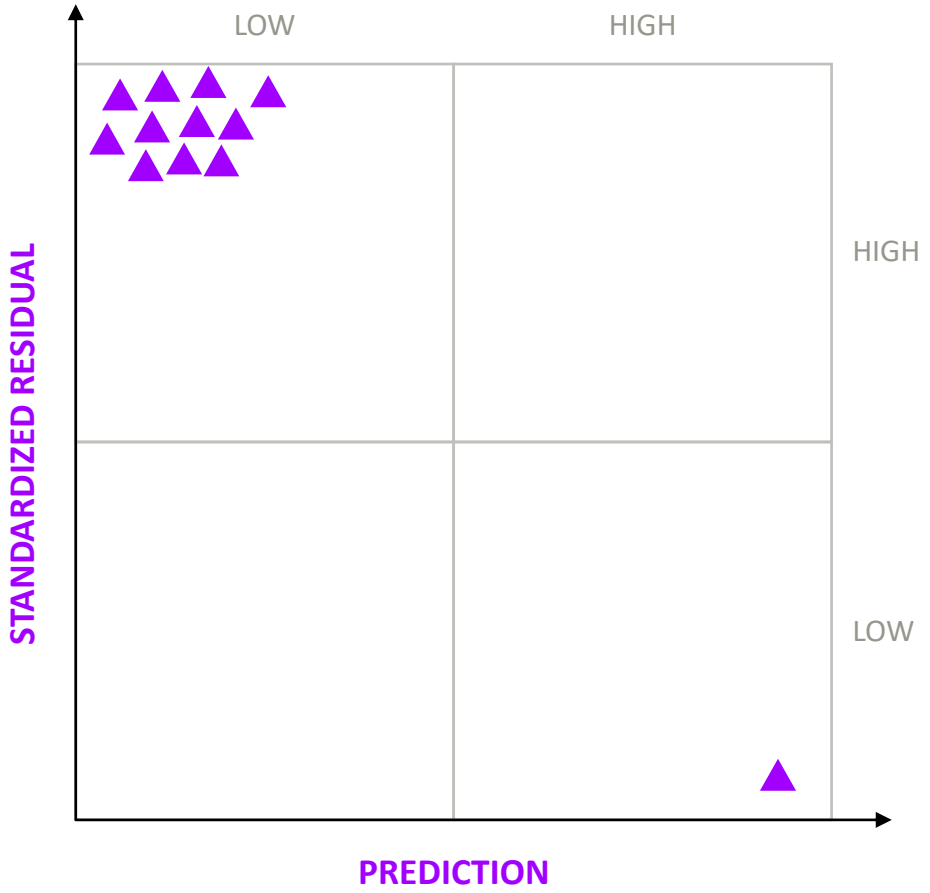
1. An indicator that your model can be improved.
2. Possibly a variable that is right on the border of significance may end up erroneously on the wrong side of that border.
3. Your regression coefficients might still be accurate.

Potential solution:

- The most frequently successful solution is to transform a variable.
- Often heteroscedasticity indicates that a variable is missing.



Outliers



Potential Solutions:

1. It's possible that this is a measurement or data entry error, where the outlier is just wrong, in which case you should delete it.
2. It's possible that what appears to be just a couple outliers is in fact a power distribution. Consider transforming the variable if one of your variables has an asymmetric distribution.
3. If it is indeed a legitimate outlier, you should assess the impact of the outlier.

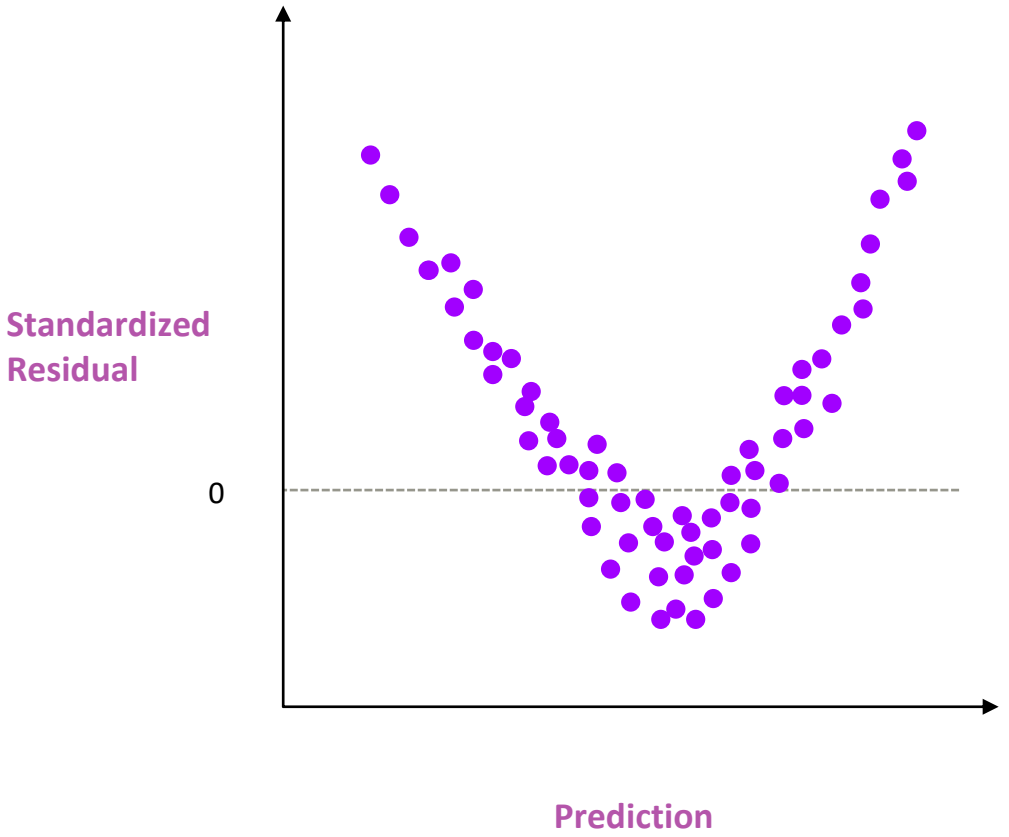
Nonlinearity

Potential Solution

IMPLICATION AND SOLUTION

Sometimes patterns like this indicate that a variable needs to be transformed.	If the pattern is actually clear, you probably need to create a nonlinear model.	It's possible that the issue is a missing variable.	There might actually not be an issue at all.

Example



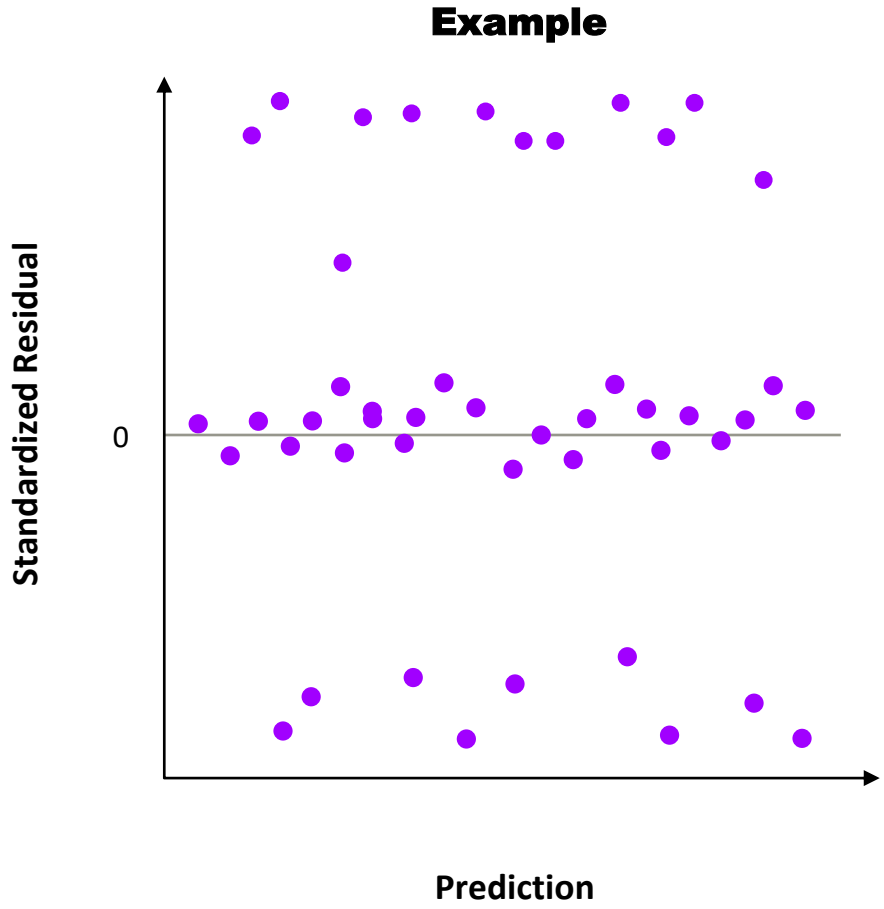
Large Y-axis Datapoints

Implication:

1. Your model isn't not as good as if you had all the variables you needed. Y
2. This model is pretty accurate most of the time, but then every once and a while it's way off.

Potential Solution:

- It's almost always worth looking around to see if there's an opportunity to usefully transform a variable.
- You probably need to deal with your missing variable problem.



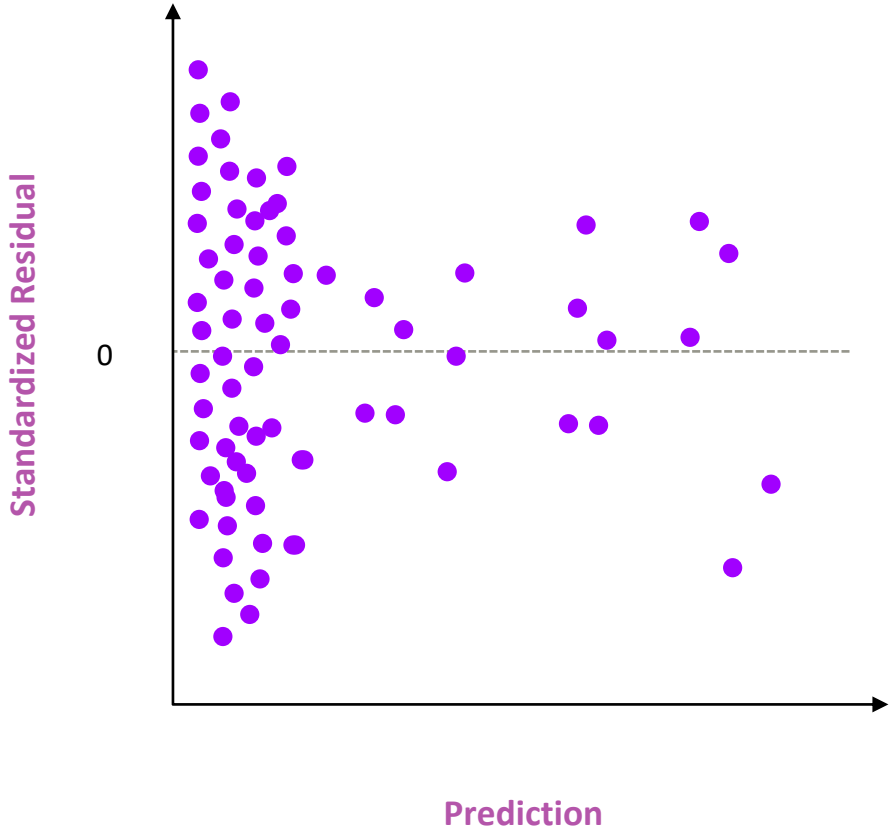
X-axis Unbalanced

Implication

- Sometimes there's actually nothing wrong with your model. In the example, it's quite clear that this isn't a good model, but sometimes the residual plot is unbalanced, and the model is quite good.
- The only ways to tell are to
  - Experiment with transforming your data and see if you can improve it and
  - Look at the predicted vs. Actual plot and see if your prediction is wildly off for a lot of datapoints.
- While there's no explicit rule that says your residual can't be unbalanced and still be accurate, it's more often the case that an x-axis unbalanced residual means your model can be made significantly more accurate.

Potential Solution

- The solution to this is almost always to transform your data, typically an explanatory variable.
- It's also possible that your model lacks a variable.



Evaluate and Improve Classification Models

Building A Spam Classifier

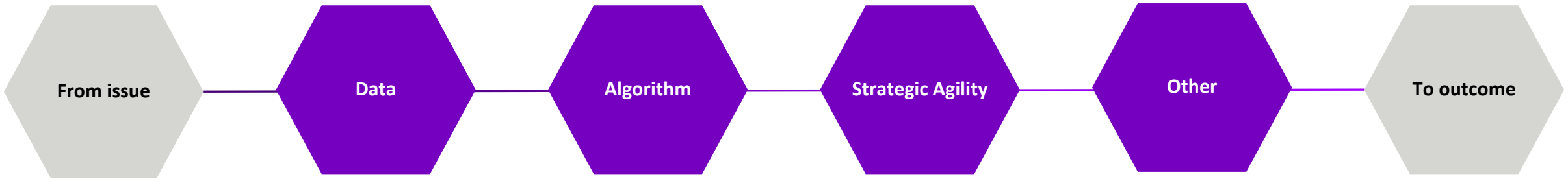
From: cheapsales@buystufffromme.com
To: ang@cs.stanford.edu
Subject: Buy now!

Deal of the week! Buy now!
Rolex w4tchs - $100
Med1cine (any kind) - $50
Also low cost M0rgages available.

From: Alfred Ng
To: ang@cs.stanford.edu
Subject: Christmas dates?

Hey Andrew,
Was talking to Mom about plans for Xmas. When do you get off work. Meet Dec 22?
Alf

Building A Spam Classifier



- How to spend your time to make it have low error?

- Collect lots of data
- Example: “honeypot” project.

- Develop sophisticated features based on email routing information (from email header).

- Develop sophisticated features for message body,
- Example: should “discount” and “discounts” be treated as the same word? How about “deal” and “Dealer”? Features about punctuation?

- Develop sophisticated algorithm to detect misspellings
- Example: m0rtgage, med1cine, w4tches.)

- High performing spam classifier

Recommended Approach

How do you begin building your model?

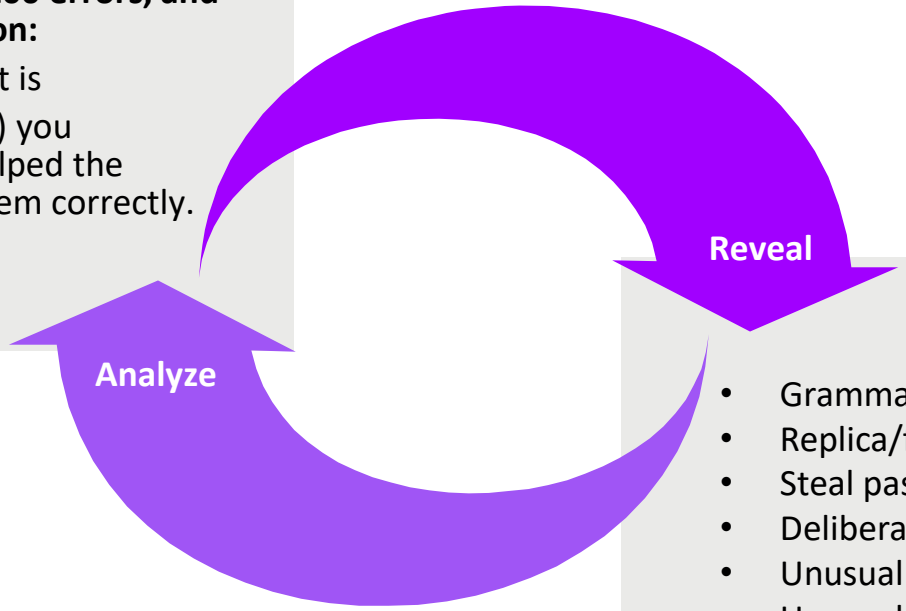
Initiation	Execution	Evaluation
Start with a simple algorithm that you can implement quickly • Implement it and test it on your cross-validation data	**Plot learning curves** • Decide if more data, more features, or other factors are likely to help	**Error analysis** • Manually examine the examples (in cross validation set) that your algorithm made errors on. • See if you spot any systematic trend in what type of examples it is making errors on.

Error Analysis

Given 500 examples in cross validation set, algorithm misclassifies 100 emails.

Manually examine the 100 errors, and categorize them based on:

- What type of email it is
- What cues (features) you think would have helped the algorithm classify them correctly.



- Grammar
- Replica/fake
- Steal passwords
- Deliberate misspellings
- Unusual email routing
- Unusual (spamming) punctuation

The Importance Of Numerical Evaluation

Mispelling	Casing	Specific Content	Unusual Markings

Should discount/discounts/discounted/discounting be treated as the same word? To solve, we can use "stemming" software

Error analysis may not be helpful for deciding if this is likely to improve performance. Only solution is to try it and see if it works.

Need numerical evaluation (e.g., cross validation error) of algorithm's performance with and without stemming

When data is large or error is too large, you need some initial metrics to know where to look

Cancer Classification Example

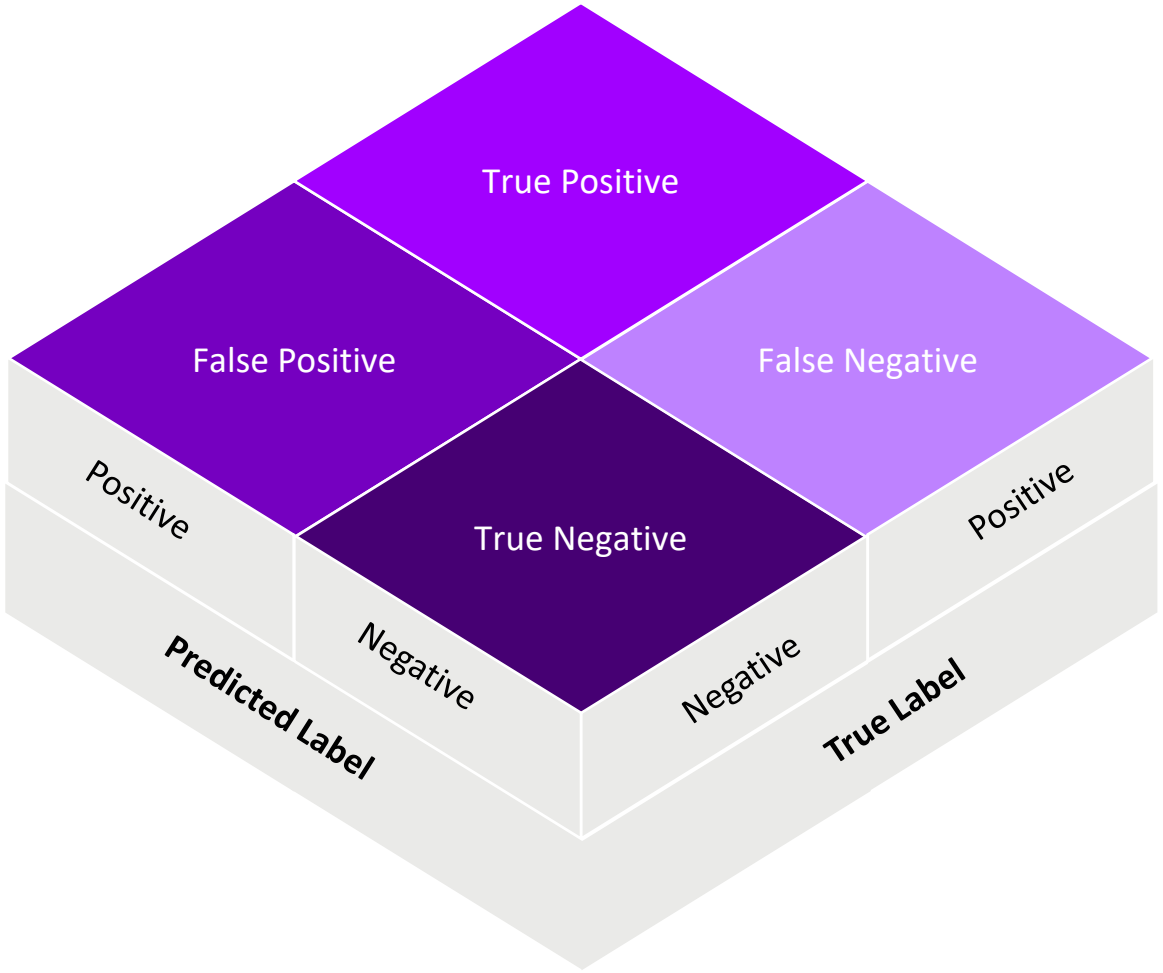
Approach:

- Train logistic regression model where you always label “negative”.
- Find that you got 1% error on test set (99% correct diagnoses)

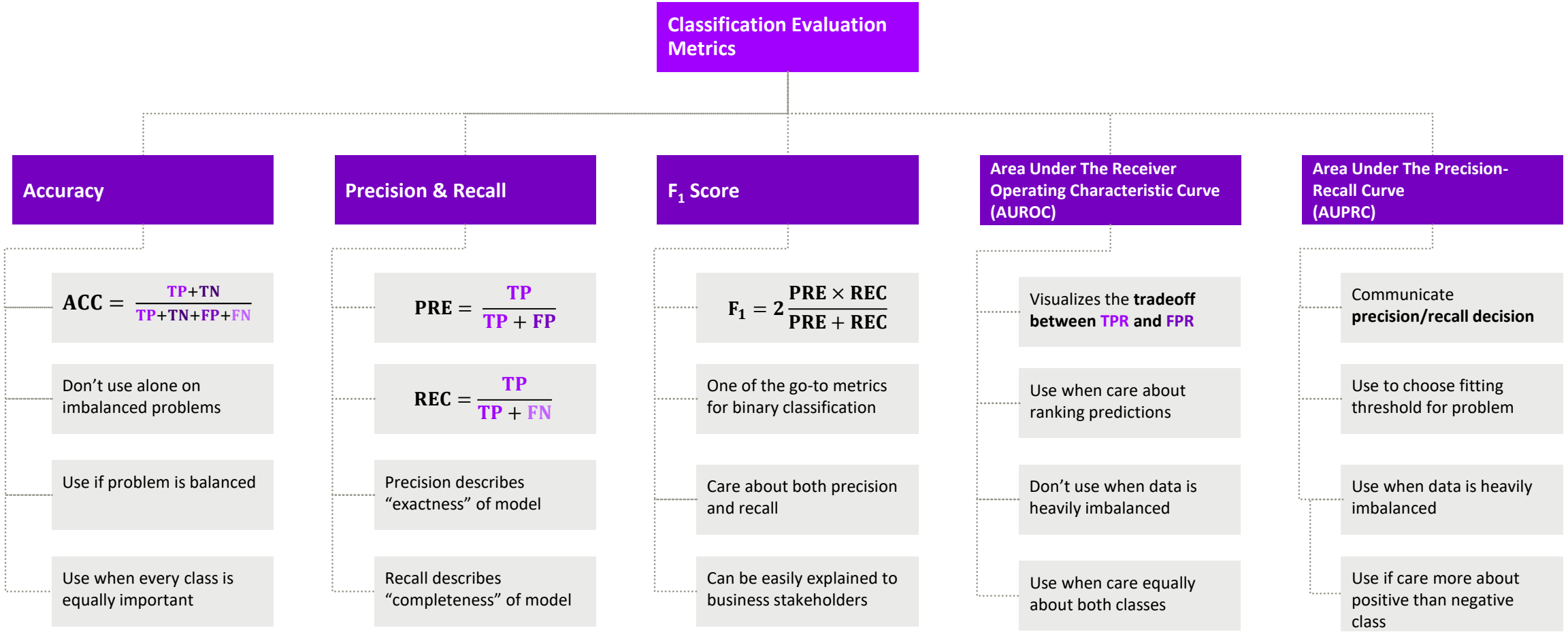
What if only 0.50% of patients have cancer?

Say you want to actually predict the “positive“ class. What do you do?

Confusion Matrix



Definitions of Important Metrics



Precision, Recall, and F_1 Score

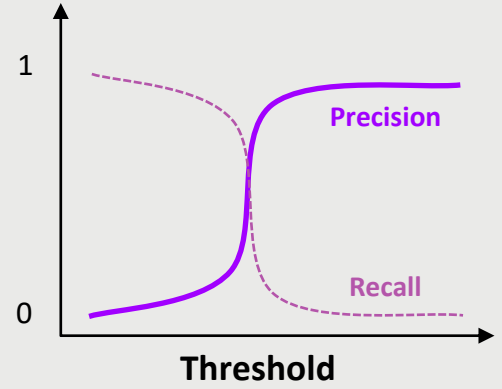
Precision

- Of all patients where we predicted, what fraction actually has cancer?
- Unlike accuracy, each class/label has a value

Recall

- Of all patients that actually have cancer, what fraction did we correctly detect as having cancer?
- Unlike accuracy, each class/label has a value

Precision-recall Trade-off



F_1 Score combines precision and recall (F_β for weighted case)

Accurary vs. AUROC

Which is better for your problem?

Accuracy

- Accuracy looks at fractions of correctly assigned positive and negative classes. That means if our problem is highly imbalanced we get a high accuracy score by simply predicting that all observations belong to the majority class.
- If your problem is balanced and you care about both positive and negative predictions, accuracy is a good choice because it is simple and easy to interpret.

AUROC

- Accuracy is based on the predicted classes while AUROC is based on predicted scores.
- Another thing to remember is that AUROC is especially good at ranking predictions.
- If you have a problem where sorting your observations is what you care about AUROC is likely what you are looking for.

Choose based on data and problem!

Accuracy vs. F_1

Accuracy

Characteristics:

- Accuracy will be high will be high even for bad models in imbalanced problems.
- If you care equally about true negatives and true positives, then accuracy is the metric you should choose.

F_1

Characteristics

- F_1 score is balancing precision and recall on the positive class while accuracy looks at correctly classified observations both positive and negative.

Verdict:

- Both of those metrics take class predictions as input so you will have to adjust the threshold regardless of which one you choose.
- F_1 is clear winner for imbalanced datasets.

AUROC vs. AUPRC

AUROC	AUPRC
• Looks at a **True Positive Rate** and **False Positive Rate**. • If you care equally about the positive and negative class or your dataset is quite balanced, then going with AUROC is a good idea. • Can give a false sense of very high performance when in fact your model can be doing not that well.	• Looks at **Positive Predictive Value** and **True Positive Rate**. • If you care more about the positive class, then using AUPRC. • It is quite common that you'd have highly imbalanced dataset where the fraction of positive class is small. Hence, AUPRC is a very useful metric in practice.

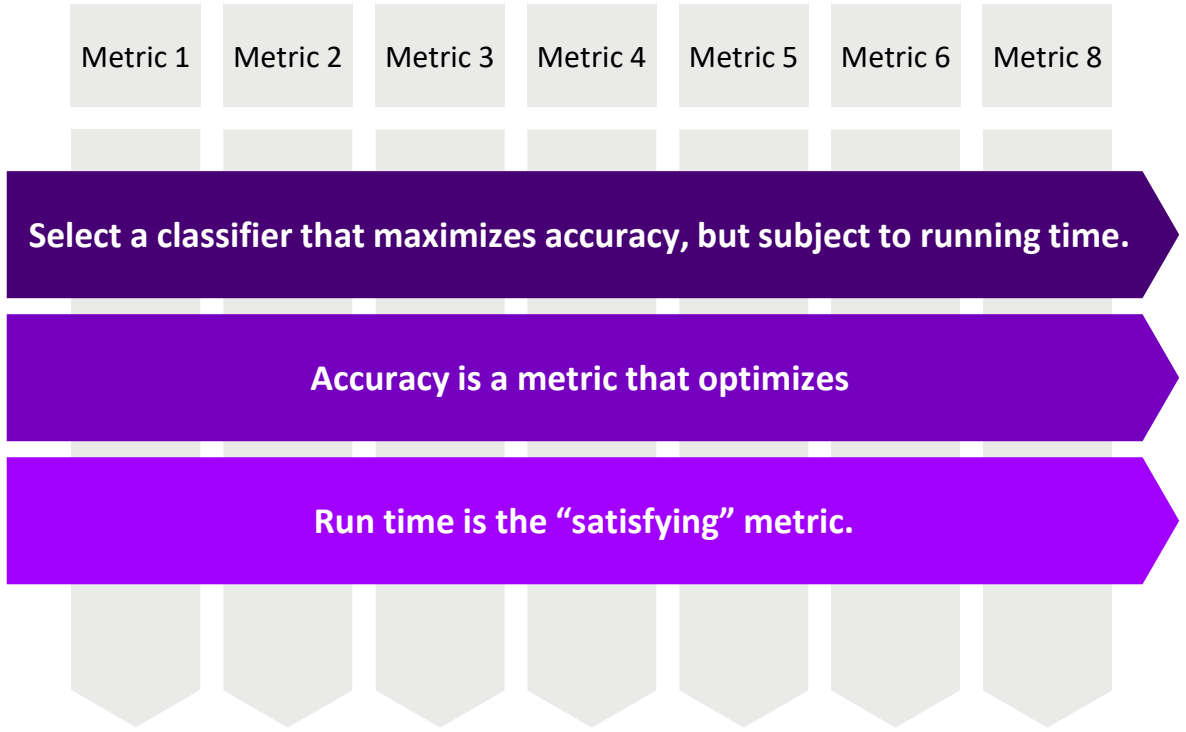
Which to use?

F_1 Score vs. AUROC

F_1 Score	AUROC
• Takes predicted classes as input • Need to choose a threshold that assigns observations to classes • Often, can improve model performance by a lot if choose well • If your dataset is heavily imbalanced and/or you mostly care about the positive class, use F_1 score, or AUPRC • Easier to interpret and communicate to business stakeholders	• Takes predicted scores as input • Use if care about ranking predictions, don't need them to be properly calibrated probabilities, and dataset is not heavily imbalanced

Optimizing Evaluation Metrics

How To Choose Your Satisficing And Optimizing Metric ?



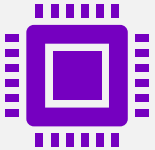
The general rule is:

$$N_{\text{metric}}: \begin{cases} 1 & \text{Optimizing} \\ \text{N}_{\text{metric}} - 1 & \text{Satisficing} \end{cases}$$

Evaluation Metrics for Other ML Problems

Notable Evaluation Metrics

BLEU Score (Precision)/ ROUGE Score (Recall): often used for text translation and summarization task.



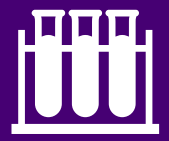
BERTscore: an automatic evaluation metric used for testing the goodness of text generation systems.

Coherence score: use in topic modeling to measure how interpretable the topics are to humans.

NDCG (Normalized Discounted Cumulative Gain): NDCG is calculated by dividing the discounted cumulative gain (DCG) of the ranked list by the DCG of the ideal ranked list, which is the list with the relevant items ranked in the most optimal order.

Evaluate Query

KPI (Key Point Indicator) measurement: an overall daily service evaluation used to measure how well our service is performing.



Validation basket: frequent testing, including but not limited to A/B experiments and fine-tuning of our service.

Test basket: a separate basket for pre-release checks that we can use to combat overfitting in the KPI basket.

Krippendorff’s Alpha

- **Krippendorff's alpha coefficient** is a statistical measure of the agreement achieved when coding a set of units of analysis.
- Used in **content analysis** where textual units are categorized by trained readers, in **counseling and survey research** where experts code open-ended interview data into analyzable terms, in **psychological testing** where alternative tests of the same phenomena need to be compared, or in **observational studies** where unstructured happenings are recorded for subsequent analysis.
- Krippendorff's alpha is applicable to any number of coders, each assigning one value to one unit of analysis, to incomplete (missing) data, to any number of values available for coding a variable, to binary, nominal, ordinal, interval, ratio, polar, and circular metrics, and it adjusts itself to small sample sizes of the reliability data.
- **The virtue of a single coefficient with these variations is that computed reliabilities are comparable across any numbers of coders, values, different metrics, and unequal sample sizes.**

The Large Data Argument

Designing A High Accuracy Learning System

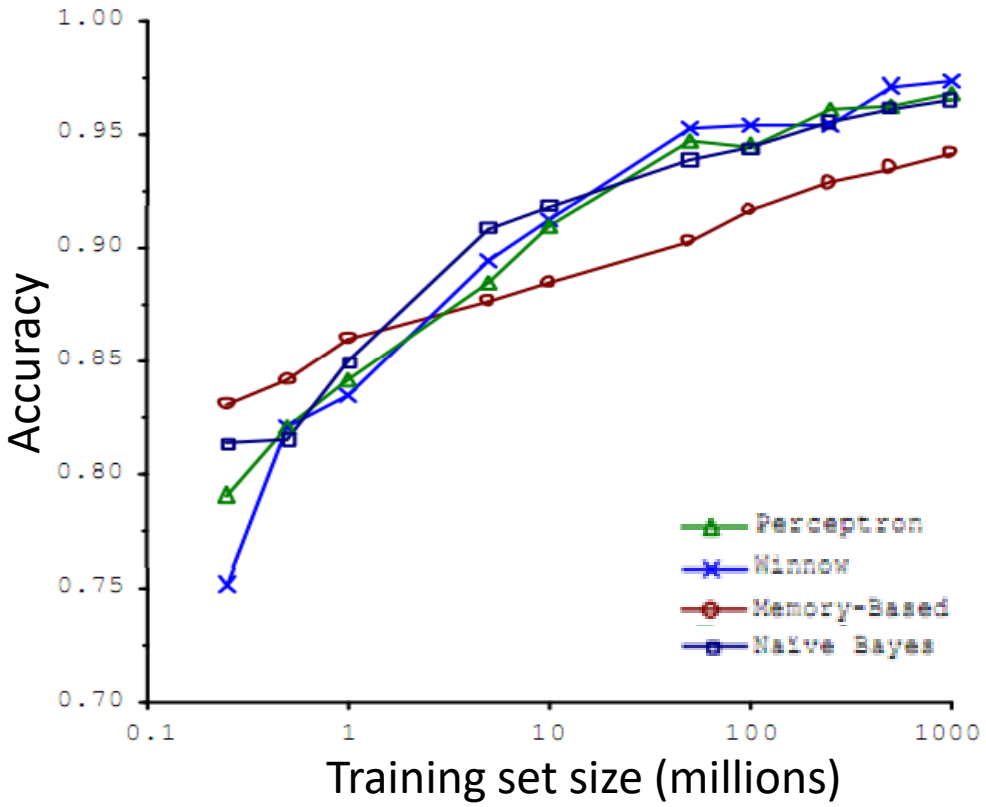
Large data rationale:

1. Use a learning algorithm with many parameters (e.g. logistic regression/linear regression with many features; neural network with many hidden units).
2. Use a very large training set (unlikely to overfit)

"It's not who has the best algorithm that wins. It's who has the most data."

Examples:

- Classify between confusable words: {to, two, too}, {then, than}
- For breakfast I ate _____ eggs.
- Algorithms
  - Perceptron (Logistic regression)
  - Winnow
  - Memory-based
  - Naïve Bayes



It’s Coding Time!

Working Session

Case study:
- Facebook ad and consumer behavior

Data:
- Source: https://www.kaggle.com/datasets/loveall/clicks-conversion-tracking

Problems:
- Read in the data and perform some exploratory analysis
- Create a new column for rate of click (Click/Impression).
- Build a predictive (regression) model to predict the rate of click using a selection of other features
    - Can you perform the train/test/validation split?
    - Can you compute the three/four types of errors we discussed today?
    - Can you look at the residual plots and diagnose the problems your model might be facing?
    - How can you make the model better?

Get started:
- Python users:
    - Install Anaconda
    - Install Python 3.7
    - Create a virtual environment:
    - *conda create -n yourenvname python=x.x anaconda*
    - Install requirements: *pandas*, *numpy*, and *sklearn* to start

  OR
    - Use Google Collab

- R users:
    - Install RStudios
    - Install requirements: *tidyverse* and *gbm* to start

Referenced Materials

1. Ng, Andrew. ”Machine Learning By Prof. Andrew Ng”, (2023), GitHub repository, https://github.com/vkosuri/CourseraMachineLearning.
2. Czakon , Jakub. “F1 Score vs ROC AUC vs Accuracy vs PR AUC: Which Evaluation Metric Should You Choose?”, MLOps Blog, 26 April 2023, https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc.
3. Upadhyay, Amit. “Precision/Recall Tradeoff”, Medium, 10 Aug 2020, https://medium.com/analytics-vidhya/precision-recall-tradeoff-79e892d43134.
4. Krippendorff's alpha. (2023, April 16). In Wikipedia. https://en.wikipedia.org/wiki/Krippendorff%27s_alpha.
5. Shankhar, Bibek Shah. “How to choose your Satisficing and optimizing metric?”, Medium, 10 Jun 2020, https://medium.com/structuring-your-machine-learning-projects/satisficing-and-optimizing-metric-24372e0a73c.
6. Ustalov, Dmitry. “Guide to Data Labeling for Search Relevance Evaluation”, Medium, 22 Apr 2022, https://towardsdatascience.com/guide-to-data-labeling-for-search-relevance-evaluation-a197862e5223.
7. Agrawal, Samarth. “How to split data into three sets (train, validation, and test) And why?”, Meidum, 17 May 2021, https://towardsdatascience.com/how-to-split-data-into-three-sets-train-validation-and-test-and-why-e50d22d3e54c.
8. Qualtrics (2023) Qualtrics XM. Available at: https://www.qualtrics.com/support/stats-iq/analyses/regression-guides/interpreting-residual-plots-improve-regression/.
9. Ahmed, Mazen. “Ways to Evaluate your Regression Model”, Meidum, 06 Oct 2021, https://linguisticmaz.medium.com/evaluating-regression-models-cb02ba075e16.

