GENERAL MODEL BUILDING NOTES

Brandon Kang
ISyE 4031: Regression & Forecasting

1.
DATA, DATA, DATA!

> “The **performance** of machine learning methods is **heavily** dependent on the **choice of data representation** (or features) on which they are applied”
> (Bengio et al., 2013)

DATA DESCRIPTION

What is your GOAL?

What data do you NEED?

How can you COLLECT it?

What is the SIZE of your data?

Is it **REPRESENTATIVE**?

HEILMEIER'S QUESTIONS

Set of questions to think through and evaluate proposed projects.

1. What are you trying to do? Articulate your objectives using absolutely no jargon.
2. How is it done today, and what are limits of current practice?
3. What is new in your approach and why do you think it will be successful?
4. Who cares? If you are successful, what difference will it make?
5. What are the risks?
6. How much will it cost?
7. How long will it take?
8. What are mid-term and final "exams" to check for success?

DATA CLEANING

60% of a data scientist’s time is devoted to **data cleaning**

Pay attention to the following…

Human error

Outliers

Text data

Missing values

Formatting

Inaccurate data

EXPLORATORY DATA ANALYSIS

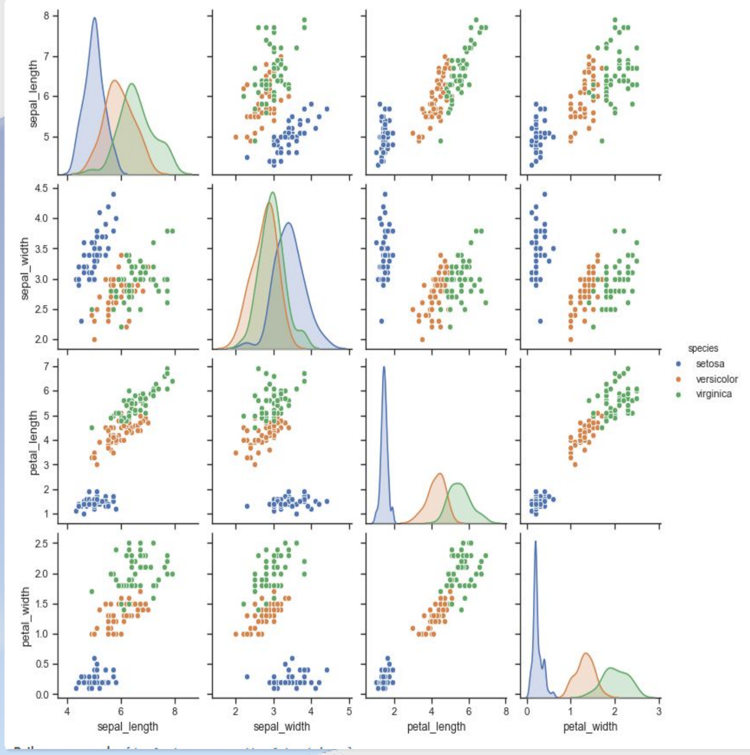
Understand your data through

1. Visualizations, such as matrix plots
2. Basic statistics (median, spread, outliers, etc.)
3. Correlations
4. Domain expertise

Visualize **relationships** between features

Understand **distribution** of features

Analyze **anomalies**, **cluster of points**, etc.



With **domain expertise** and **exploratory data analysis**, grasp **WHY** relationships and patterns exist.

Use Machine Learning to discover hidden patterns, but don’t rely on ML -- your **initial analysis** is often **most powerful**!

FEATURE ENGINEERING

The quality of your **model** is as good as the quality of your **features**

Can you engineer new features using raw data?

What did you discover from your initial analysis?

EXAMPLES OF FEATURE ENGINEERING

Imputing missing data

- Using mean or a rule-based approach?

Encoding/grouping categoricals

- From month data, should you combine months into seasons instead?

Standardization

- Does your algorithm require you to standardize?

Transforming

- Is your data heavily skewed?

2.
MODEL BUILDING

GENERAL PROCESS FLOW

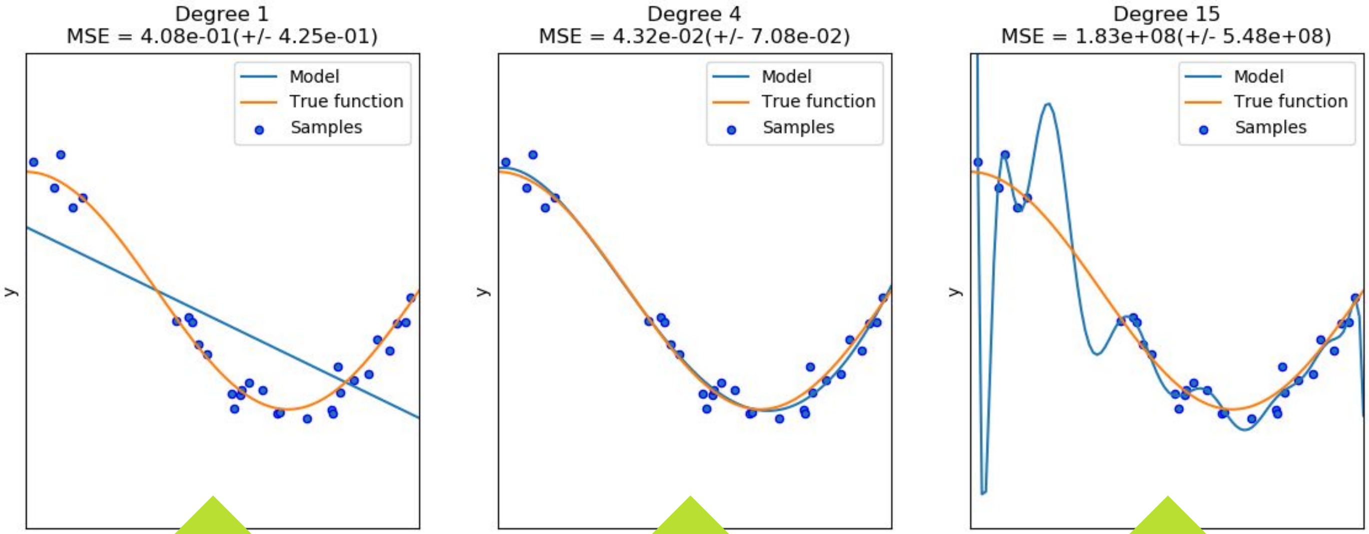
1. **Split** data into training (~80%) and testing (~20%)
2. **Build** model using training set
3. **Assess** predictive power (RMSE, MAE, etc.) with testing set

We need to split our data into training and testing to accurately assess the **predictive power** of a model.

UNDERFITTING VS. OVERFITTING

Underfitting: when your model does not capture the underlying pattern in your data

Overfitting: when your model may have captured the noise of the data; can not generalize well on new data

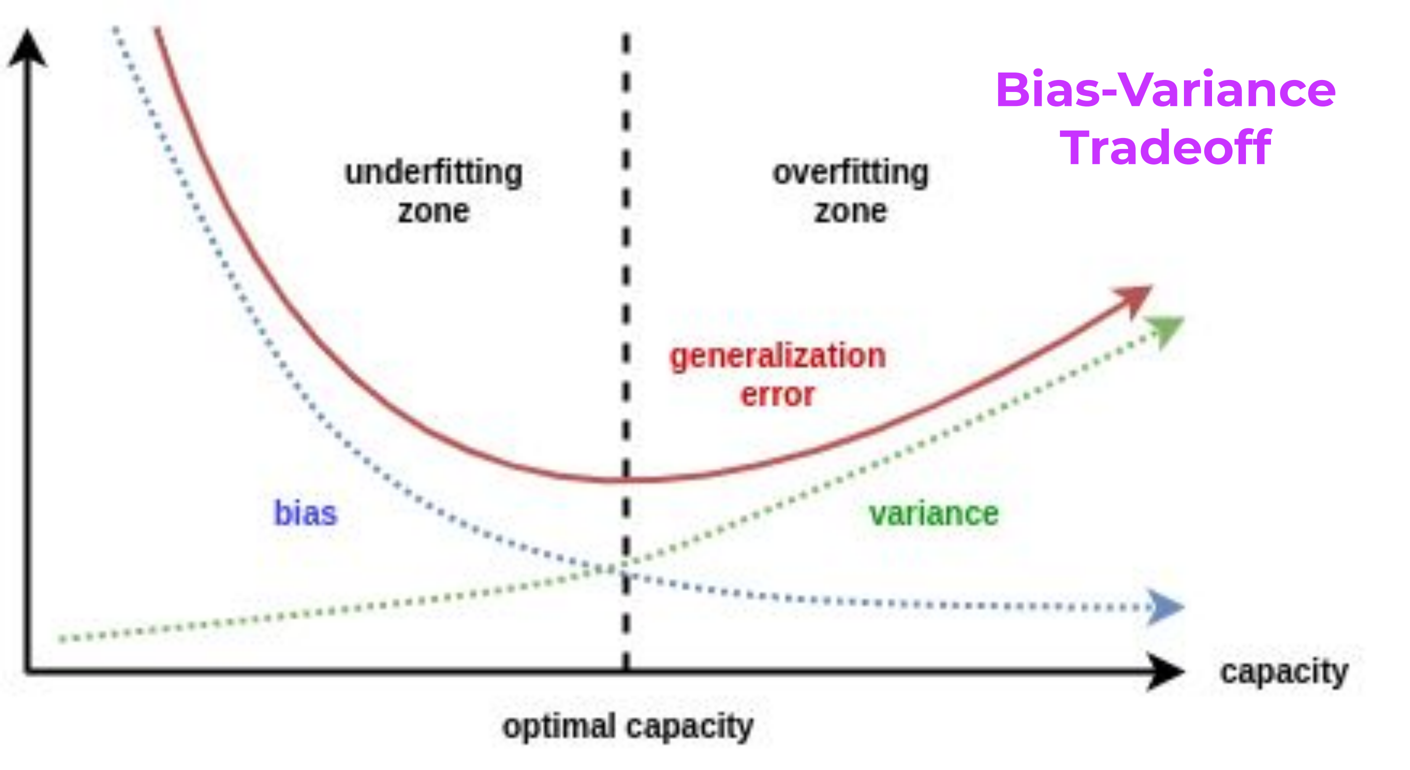


Underfitting
Low variance
High bias

Just right!

Overfitting
High variance
Low bias

Bias-Variance Tradeoff



2.1 UNDERFITTING AND OPTIMIZING PREDICTIVE POWER

STANDARD CHECKS

1. Multicollinearity
2. Outliers
   a. Does it make sense to remove all outliers in the scope of your project?
   b. **WHY** are they outliers?
3. Assumption Checking
   a. Transformations
   b. Higher order/interaction terms

AVOIDING UNDERFITTING

1. Add more parameters/higher degree terms
2. Find more relevant features if your feature space is small
3. Increase complexity or change type of model
4. Increase training time until cost function converges

OPTIMIZING PERFORMANCE

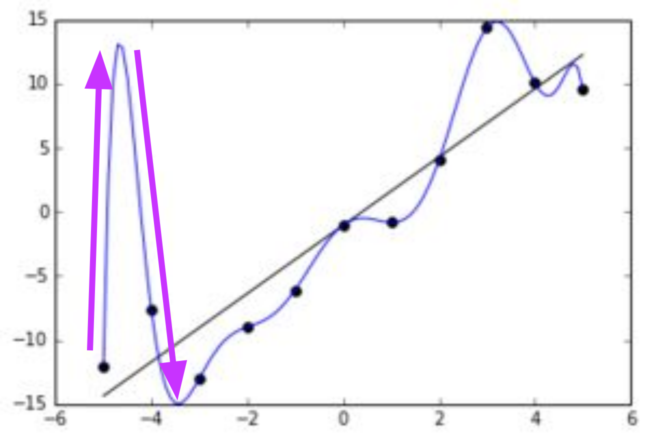
1. Multicollinearity and overfitting?
   a. Ridge Regression for multicollinearity
   b. Lasso Regression for feature selection
2. Try non-parametric models
   a. Local regression (LOESS)
   b. Gradient Boosting/Random Forest
3. Heavy influence from outliers?
   a. Robust regression
4. Only care about performance?
   a. Deep learning

All of these methods have caveats and perform better in certain situations. Understand **WHEN** to use them! Some models are harder to interpret than others (e.g neural nets, LOESS).

Everything depends on your **DATA**! Therefore, you **MUST** understand your data as best as possible!

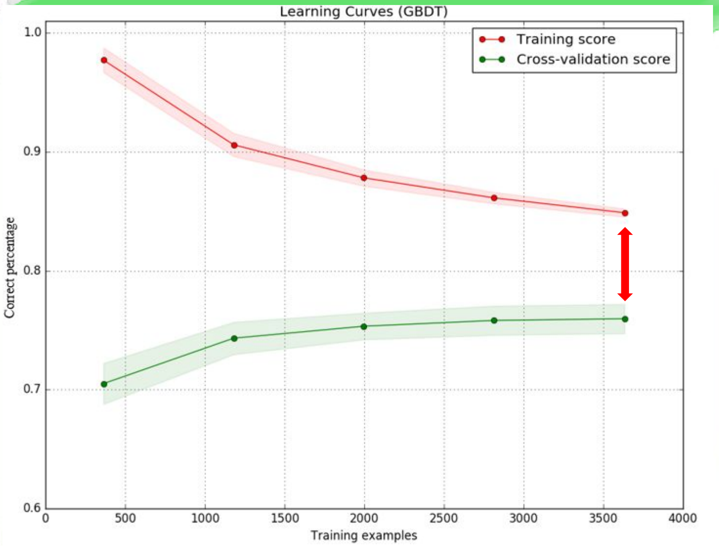
2.2
OVERFITTING

Large beta coefficients: small changes in input can cause drastic changes in output value



Metrics in training are deceptively good but very poor in validation

Use **learning curves** to assess overfitting



Validation score has converged, but still **much worse** than training score

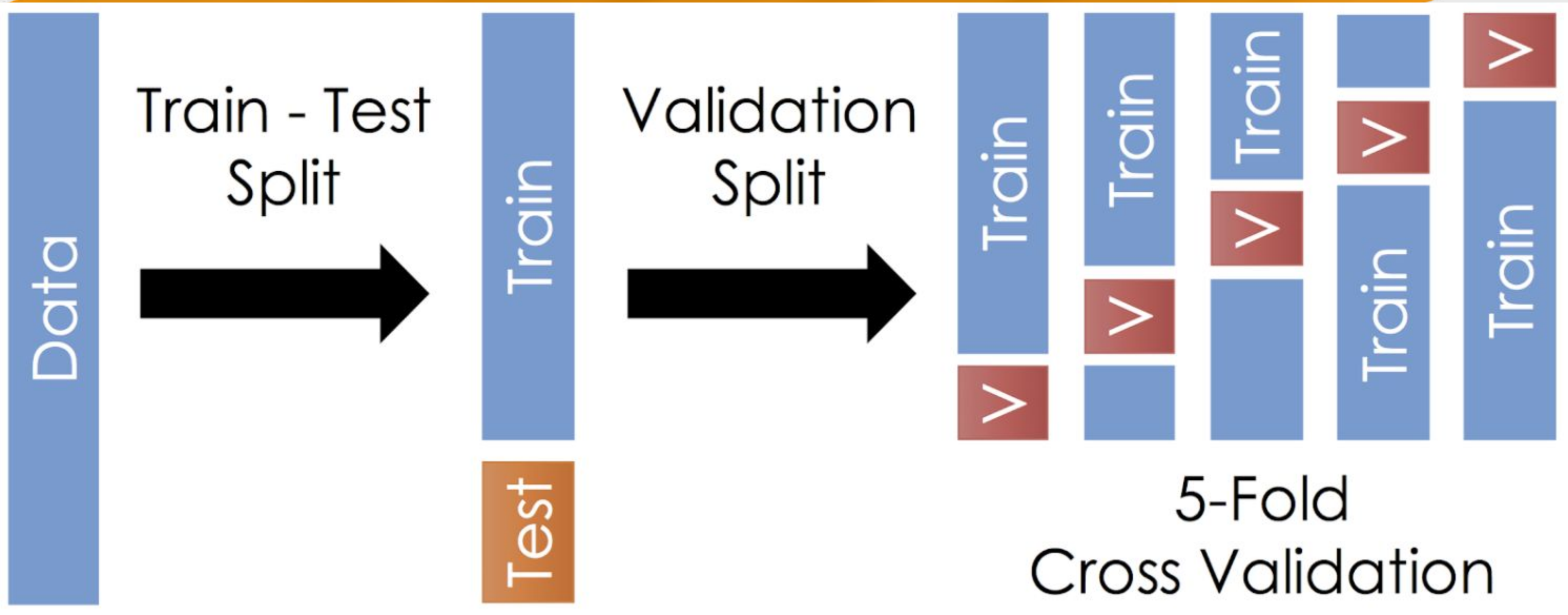
FEATURE DIMENSIONS

As the number of features grows, we need exponentially more data to generalize accurately (**curse of dimensionality**)

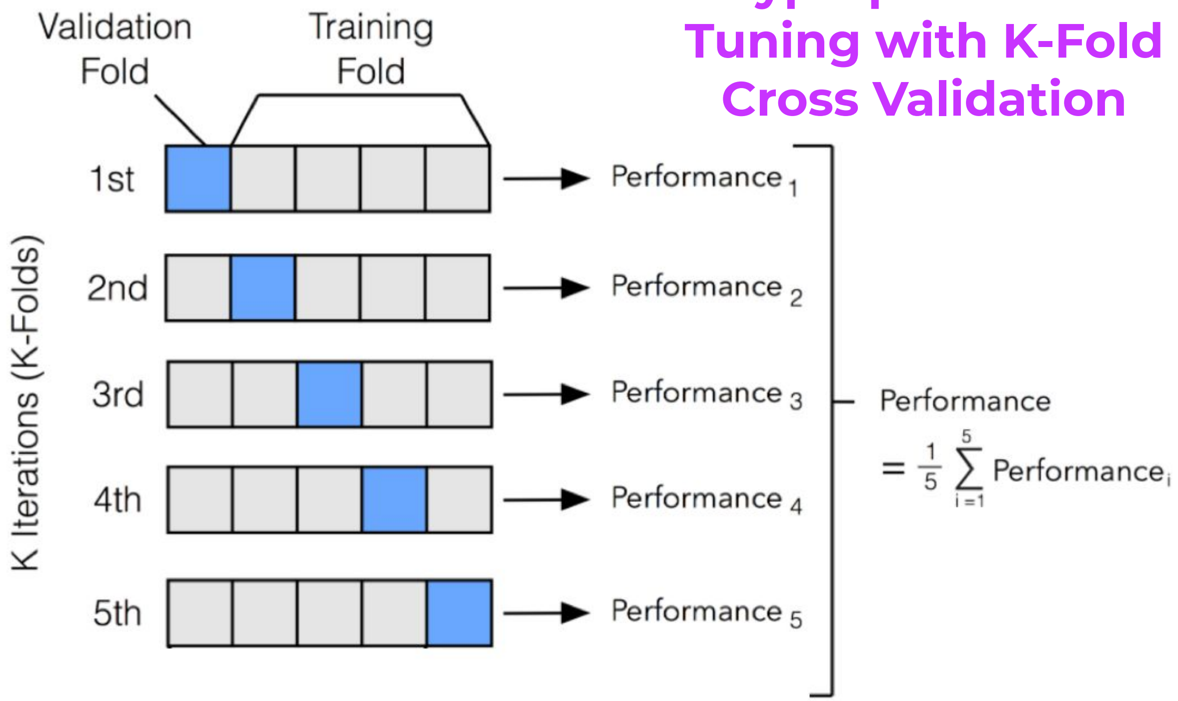
Too many features relative to the number of data points can result in **overfitting**

AVOIDING OVERFITTING

1. Feature selection methods
2. Collect more data (*not always viable)*
3. Use regularization (Ridge/Lasso)
4. Tune hyperparameters with cross validation and use early stopping for tree-based models


Data
Train - Test Split
Train
Test
Validation Split
Train
V
Train
V
Train
V
V
Train
V
Train
5-Fold Cross Validation

Hyperparameter Tuning with K-Fold Cross Validation



$$\text{Performance} = \frac{1}{5}\sum_{i=1}^{5}\text{Performance}_i$$

3. MODEL COMPARISON & PREDICTION

INTERPRETATION

Compare model performance on testing set

- Define your metrics: MSE, R^2, AIC/BIC, etc.

Do you need a model you can **interpret** or are you optimizing for **performance**?

- In a business setting, interpretation may be more powerful (i.e deep learning isn’t always the solution!)