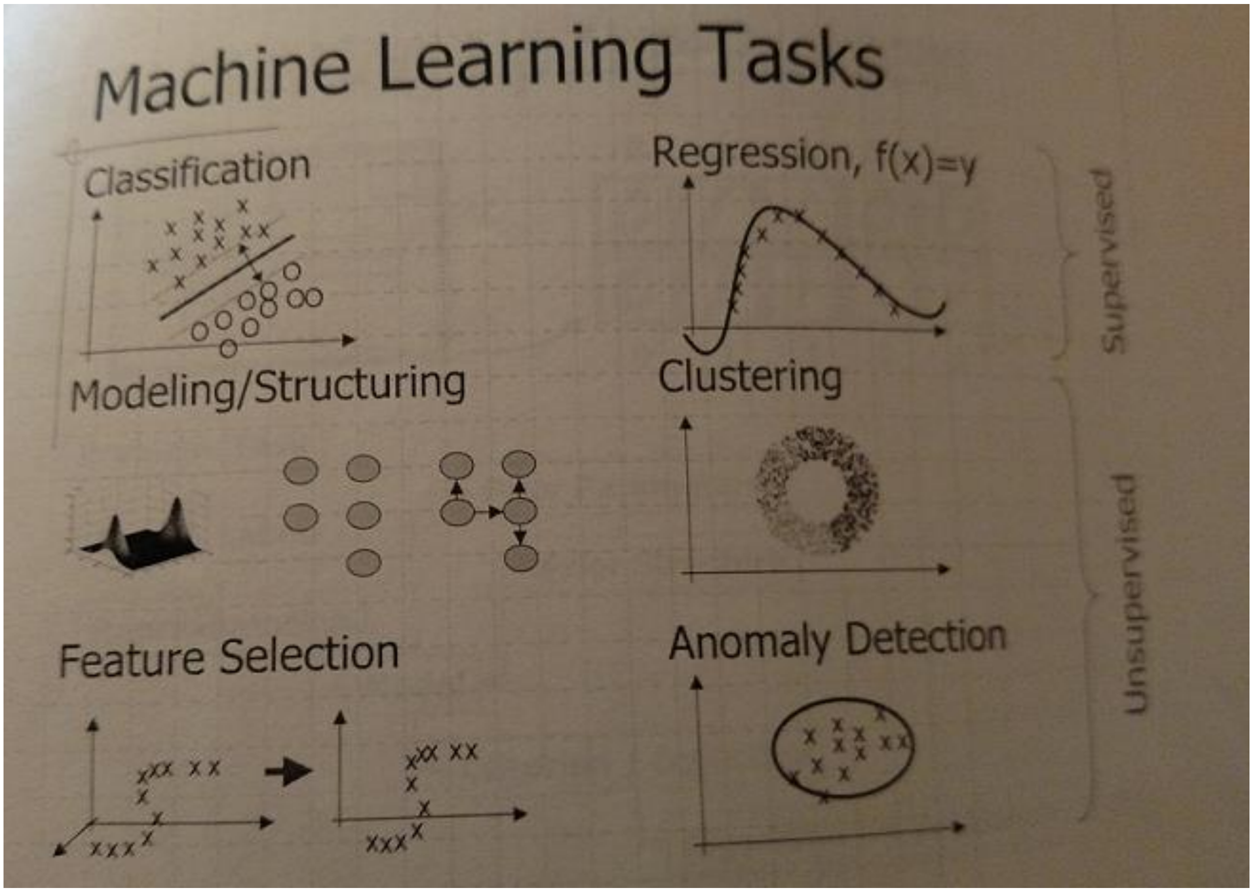

Machine Learning Tasks
Classification
Regression, f(x)=y
Supervised
Modeling/Structuring
Clustering
Feature Selection
Anomaly Detection
Unsupervised

Supervised vs. Unsupervised Learning

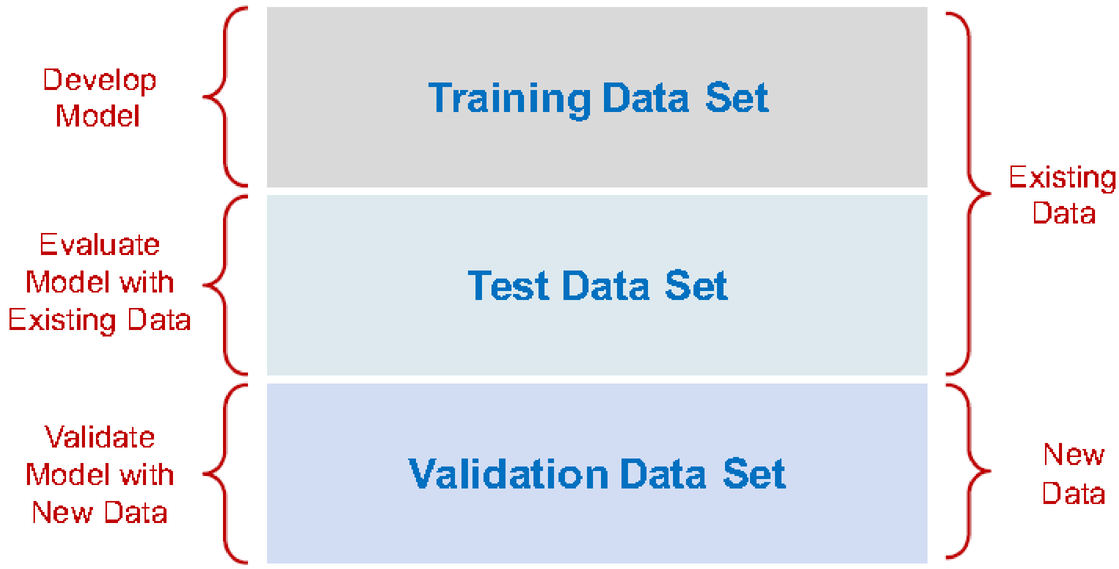
- Machine learning can be **supervised** or **unsupervised**.
- We learn many things in life (e.g., **how to walk**) without even thinking or **without** specific **goals** in mind.
- Likewise, when we apply machine learning methods to learn from the data without a specific goal, this is called **"unsupervised"** learning.
- **Data mining** is ***"the computational process of discovering trends in data (ACM)"*** which were previously unknown – i.e., more closely associated with **"unsupervised learning"**.
- When you explore **descriptive statistics** and **correlations** you are using unsupervised learning methods.
- We learn other things in life with a **specific purpose** or **goal** (e.g., predictive analytics).
- Likewise, when we apply machine learning methods with a specific goal in mind, we call this **"supervised"** learning
- Most **predictive analytics** methods fall under the category of **"supervised"** learning – there is a specific **prediction goal**

KOGOD SCHOOL *of* BUSINESS
AMERICAN UNIVERSITY • WASHINGTON, DC

Key Machine Learning Concepts

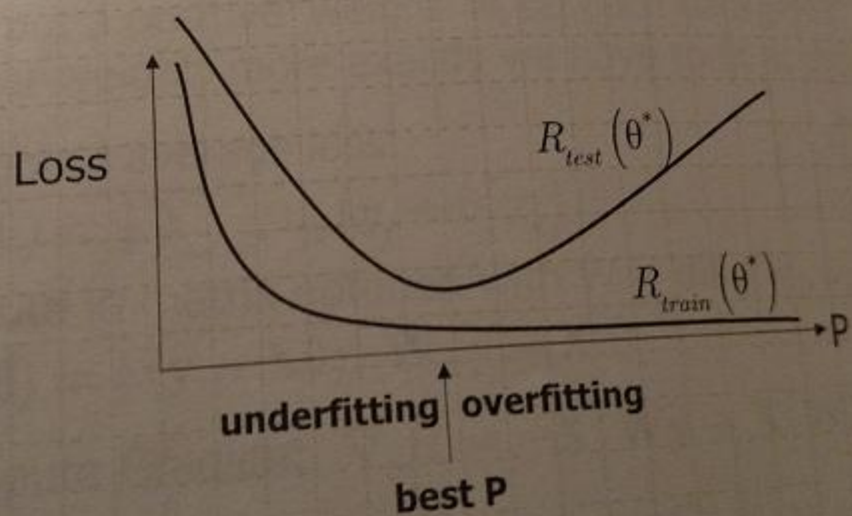
- **Training** – when we **run models** on the data to obtain parameters to help us understand the data (e.g., means, variance, regression coefficients), this is called "training" the model.
- **Testing** – some time we use all the data to develop predictive models, but this is not very useful for evaluating the predictive accuracy of the data. So, it is customary to set aside a portion of the data to test the model.
- **Training/Test/Validation Data Sets** – when the data is **partitioned** into a part to train the model and the other part to test the model, we referred to these data portions as the **training** and **test** data **sets**. When we obtain new data to evaluate the model we call this new data the **"validation"** data set.
- This is particularly important because **over-identified** models are notorious for fitting the existing data well, but **performing poorly** with different data
- Partitioning the data into various training and test sets and computing aggregate predictive accuracy scores is **central to machine learning**.

Machine Learning Illustration



Crossvalidation

- Try fitting with different polynomial order P
- Select P which gives lowest $R_{test}(\theta^*)$



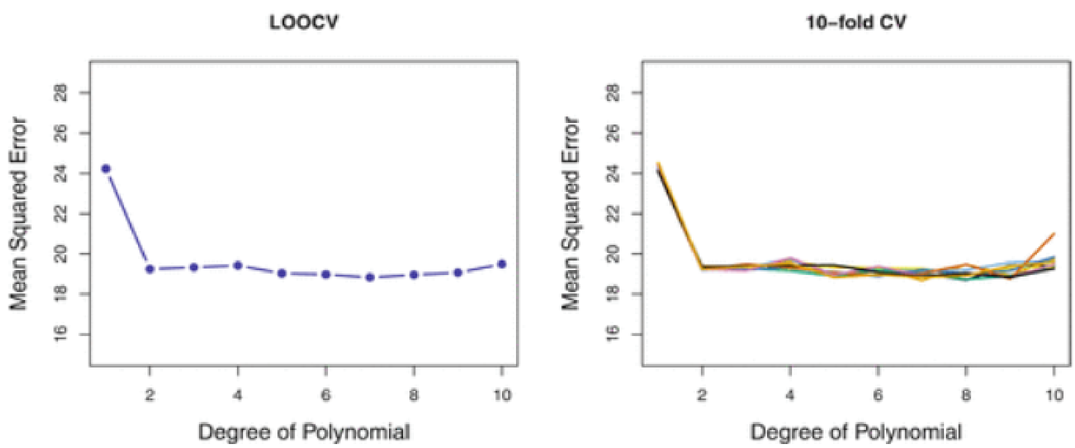
Partitioning and Re-Sampling

- **Re-sampling** involve drawing **samples** from the data many times and re-fitting (i.e., **re-training**) the model each time to test the model more thoroughly.
- There are **many ways** to **partition** the data when sampling and re-sampling data.
- Most popular partitioning **methods** include:
  - **Hold-out** random splitting (**pre-set** percentage).
  - **K-Fold**
  - **Leave-One (or P) Out**
  - **Bootstrap**

KOGOD SCHOOL of BUSINESS
AMERICAN UNIVERSITY • WASHINGTON, DC

LOOCV vs. 10-Fold CV

The example below was generated with the **"Auto"** data set in R, predicting **gas mileage** with **horsepower** (same model we showed before). The two graphs show that the LOOCV and the 10-Fold CV **performed similarly**, but the 10-Fold CV only requires 10 regression model estimations, whereas LOOCV requires one for each data point.

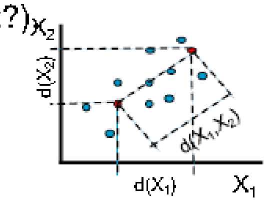


Bootstrap

- The **Bootstrap** method is similar to **cross-validation**
- Like with cross-validation, the model is **trained** with samples **selected** from the data and **tested** with the observations that were **not selected**
- The **difference** is that in the Bootstrap method:
  - **S sample** observations are drawn **K times**
  - Each of the **K samples** is done **"with replacement"**, which means that the same data can be re-selected
  - While this may seem redundant, each time you draw a new sample, each data point has the **same probability** of being selected
  - Also, there is **no limit** to how many times samples can be drawn
  - So, if a data set has **N observations**, one could draw **N** different random **samples**, thus having as many samples as data points.
  - Bootstrapping is **popular** when the **distribution** of the data is **unknown** or has unusual shapes because the means of the samples extracted are approximately normally distributed

Dimensionality Problems

- **Multi-Collinearity:** high correlation between independent variables cause the model to be unstable (e.g., dropping a few data points may yield substantially different results).
- **Over-Identification:** more variables force the model to fit the data tighter, but this is no guarantee that the model will make accurate predictions for new data.
- **Less Degrees of Freedom:** every added variable reduces the degrees of freedom of a model (n-p-1).
- **Less Parsimony:** complex models difficult to interpret. Some variables will be highly significant, others not so much (keep them or not?)
- **High Variance:** while adding more variables to a model reduces bias, the additional dimensions increase the variance of the model because the distance between points becomes larger



- **Nuisance (or Noise) Variables:** adding variables that are not very relevant for the model distorts its predictive accuracy and increases variance

In a nutshell, how many variables to include in the model is a **tradeoff!!**

Addressing Dimensionality Issues

- There are a number of modeling **techniques** to deal with high dimensionality. The most popular types are:
  - ✓ **Variable Selection** – if there are too many variables in the model, the most obvious solution is to carefully **select** which ones to include or not and testing the resulting models
  - ✓ **Shrinkage or Regularization** – when business rationale suggests that all or many available variables should be included in the model, dimensionality problems can be minimized by assigning **low weight** to unimportant variables by **shrinking** their **coefficients**, rather than removing them all together.
  - ✓ **Dimension Reduction Methods** – variables can be **grouped** and **combined** into fewer (i.e., reduced) **components**
  - ✓ **Structural Equations** – estimation is done with two or more **related models**, rather than a single model – i.e., a dependent variable in one model can be an independent variable in another model (covered later in the semester)

Testing for Multi-Collinearity

- First, you need to analyze the **correlation matrix** and inspect for **desirable** correlations → **high** between the **dependent** and any **independent** variable; and **low** among **independent** variables.
- Run your regression model and report **multi-collinearity statistics** in the results. Two are most widely used:
  - **Condition Index (CI):** a composite score of the linear association of all independent variables for the **model** as a **whole**
    - ✓ **Rule of thumb: CI < 30** no problem, **30 < CI < 50** some concern, **CI > 50 severe**, no good
  - **Variance Inflation Factors (VIF):** a statistic measuring the contribution of **each variable** to the model's multicollinearity → helps figure out which variables are problematic
    - ✓ **Rule of thumb: VIF < 10** no problem, **VIF >= 10** too high,

Subset Selection Methods

- **Full vs. Reduced** model testing – use a **business criteria** for variable selection and try a few models that make business sense
- **Best Subset Selection** – with P possible candidate variables, build **P simple** regression models, one for each variable; then build all possible regressions with **2 variables**; then 3 and so on. Use cross-validation to select the best model
  - If P is large → **combinatorial explosion** of models to test
- **Step Methods** – progressively **adding (Forward) or removing (Backwards)** variables, or **both** back and forth **(Stepwise)**

Error Measures and Model Size

- The **MSE** (all models) and **R^2** (some models) are good measures of **model fit** and individual **model quality** → **low MSE** and **high R^2**
- However, the **MSE** goes **down** and the **R^2** goes **up** as more variables are added to the model, so these are **not** so **useful** to **compare** models
- In addition, the **training MSE** tends to **underestimates** the **test MSE**, particularly as the model increases in **size** and **complexity**
- So, is it **worth** the added model **complexity** to improve the **MSE**?
- There are some measurement methods that **adjust** for the number of **variables** in a model: Mallow's **Cp,** Akaike Information Criterion **(AIC),** Bayesian Information Criterion **(BIC)** and **Adjusted R^2** (already covered)

Plots of Cp, BIC and Adjusted R2 against the number of variables for the Credit data in the ISLR package