95-891: Introduction to Artificial Intelligence

Session 3: Machine Learning and Supervised Methods

David Steier

steier@andrew.cmu.edu

September 2, 2025

Agenda

- AI applications in the enterprise
- Ideas for AI applications
- What is machine learning?
- Supervised versus unsupervised learning
- Regression – linear, ridge, lasso
- Classification – logistic regression, decision trees

Impact of AI in the Private Sector

- **Operational efficiency**
  - Cost reduction
  - Productivity improvement
- **Enhancement of decision-making**
  - Data-driven insights from predictive and prescriptive analytics
  - Real-time decision support (dashboards, anomaly detection)
- **Driving innovation**
  - Product and service development
  - Process innovation
- **Market disruption**
  - Creation of new business models
  - Industry transformation

Impacts of AI at Amazon

- AI applications at Amazon
  - Personalization algorithms driving product recommendations
  - Dynamic pricing
  - AI in logistics and supply chain optimization
  - Amazon Web Services (AWS) offering AI services to customers
- Strategic impact of AI
  - Increased customer engagement and sales.
  - Operational efficiency leading to faster delivery times.
  - Positioning as a leader in cloud-based AI solutions

AI Use Cases In the Private Sector

- **Manufacturing**
  - Predictive maintenance
  - Part inspection
- **Healthcare**
  - Diagnostics
  - Drug discovery
- **Automotive**
  - Autonomous vehicles and Advanced Driver-Assistance Systems (ADAS)
  - Warranty analytics
- **Media and Entertainment**
  - Forecasting viewership
  - Games with AI agents
- **Financial Services**
  - Risk assessment
  - Fraud detection
- **Energy**
  - Operation and maintenance of power plants and grids
  - Forecasting demand and production, especially renewable energy
- **Agriculture**
  - Precision farming
  - Optimizing yield and irrigation
- **Information technology**
  - Software design, coding, debugging
  - Cybersecurity

AI Use Cases in the Public Sector

- **Citizens/constituents:** Behavior, requirements, across different demographic/service groups
- **Transparency:** Improved public awareness of public spending and policy effectiveness
- **Commercial value:** Open up new revenue streams by licensing government data
- **Procurement analytics:** Risk-based maintenance and regulation
- **Technology:** Performance cost, utility, interoperability, and technical problems
- **Service demand:** Supply and demand projections, costs, aggregation across silos for new system understanding (e.g. social care)
- **Workforce analytics:** Absenteeism, recruitment, performance, training and development, staff unit cost, and individual capabilities, especially in light of aging workforce
- **Financial performance:** Real time balance sheet and resource account data
- **Asset analytics:** Equipment, land and buildings: value, depreciation, location, utilization rates
- **Fraud, error, and waste:** Frequency, cost, most problematic areas, benchmarking
- **Market failure:** Understanding threats and risks, tracking outcomes
- **Regulatory scrutiny:** Asking the right questions, performance tracking, driving public sector responsiveness

AI Can Be Used to Promote Social Good

- **Addressing societal issues:** AI can be used to analyze data and identify patterns of discrimination in areas like housing, lending, and criminal justice. This can inform efforts to create more equitable policies and practices.
- **Healthcare:** AI has the potential to personalize medicine and improve healthcare outcomes for everyone
- **Education:** AI-powered tools can personalize learning experiences and identify students who need extra support.
- **Economic opportunities:** AI is creating new jobs in fields like tech and data science. With proper training and access, marginalized groups can participate in this growing sector.

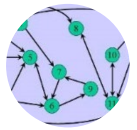
AI Final Projects From Previous Semesters

- Diagnosing patients from respiratory sounds
- Recognizing human activities from smartphone inertial measurements
- Deciding on treatment options for diabetic macular edema
- Using AI to generate paintings
- Identifying hate and offensive speech in social media posts
- Predicting song popularity from Spotify data
- Detection of oil sheen (indicating leaks) from drilling videos
- Recognizing disease in crops

Questions to Ask About Proposed AI Projects

- What exactly is the problem being solved?
- Who are you solving the problem for?
- How are you measuring solution quality?
- How good does the solution have to be to justify the project?
- What data sources will be useful in solving this problem?
- Are those data sources accessible to you?
- Are there any privacy, ethical, or legal factors to consider?
- How good are the data available to you?
- How much data do you have to analyze to solve this problem?
- How quickly do you need answers after seeing new input data?
- How long do you have for this project?
- What is your budget for people, hardware, software, data, training, etc.?
- Who is available to work on this project and what skills do they have?
- Who else has worked on this problem and how can you build on their efforts?

Techniques for Finding Patterns in Data

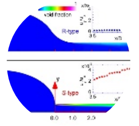


Optimization

Deterministic (LP, IP, MINLP, etc.)

Stochastic (Markov process models, queuing models)

Multi-attribute utility models

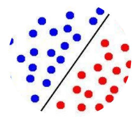


Simulations

Monte Carlo

System dynamics

Discrete event simulation



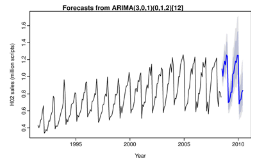
Machine Learning

Supervised

Unsupervised

Semi-supervised

Reinforcement learning



Time Series

ARIMA

ARCH/GARCH

Bayesian forecasting

Unstructured data analytics

Natural language processing

Audio/video analytics

Graph analytics

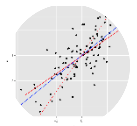
Geospatial analytics

Web analytics

Deep Learning

Recurrent neural nets,
Convolutional Neural Nets
Transformers…

Supervised Learning



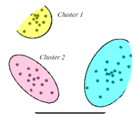
Decision Trees

Random Forests

Logistic Regression

K- NN

Unsupervised Learning



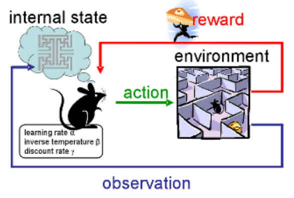
Matrix factorization

K- Means

Hierarchical clustering

Expectation maximization

Reinforcement learning



Multi-armed banded

Temporal difference

Deep=-Q-networks

Natural language processing

TF/IDF
Latent Dirichlet Allocation
Word embeddings
Large language models

Machine Learning

- **Traditional programs**: Told what to do at each step
- **Expert systems**: Given the knowledge to select appropriate actions, or search for the right action
- **Machine learning**: Allow computers to acquire knowledge from data and experience, both for performing new tasks and improving performance on previously acquired tasks
    - **Unsupervised machine learning**: Not told the right answer
    - **Supervised machine learning**: Supervisor gives feedback (or labels) with the right answer for each example

Supervised vs Unsupervised learning

- Learning from data is a process of generalization from observation
  - Ex. 1: Precipitation = 7, Wind speed = 40, Temperature = 20, Weather = Severe
  - Ex. 2: Precipitation = 0, Wind speed = 2, Temperature = 45, Weather = Mild
  - Ex. 3: Precipitation = .7, Wind speed = 9, Temperature = 30, Weather = Moderate
- **Supervised learning**: Predict the target variable by generalizing from the features associated with each observation
  - Here Weather is the target variable, while Precipitation, Wind speed and Temperature are features
  - "If Precipitation and Wind speed are high and Temperature is low, then the weather is Severe"
- **Unsupervised learning**: Find interesting groupings of the observations without using labels
  - Here just Precipitation, Wind Speed and Temperature are used
  - "Precipitation and Wind speed seem to be related"

Supervised Learning: Classification vs. Regression

- $\boldsymbol{X} = (x_1, x_2)$ and $y = \{label\ 1, \ldots, label\ n\}$ or $y \in \mathbb{R}$
- A training set with many examples of $(\boldsymbol{X}, y)$
- The model learns on the examples of the training set to produce the right value of y for an input vector $\boldsymbol{X}$

Classification

y = {yellow, gray}

y = {churn, no churn}

y = {increase, unchanged, decrease}

y = {blonde, gray, brown, red, black}

y = {job 1, job 2, ... , job n}

Numerical Predictions (Regression)

y = temperature

y = number of visitors

y = number of kW

y = price

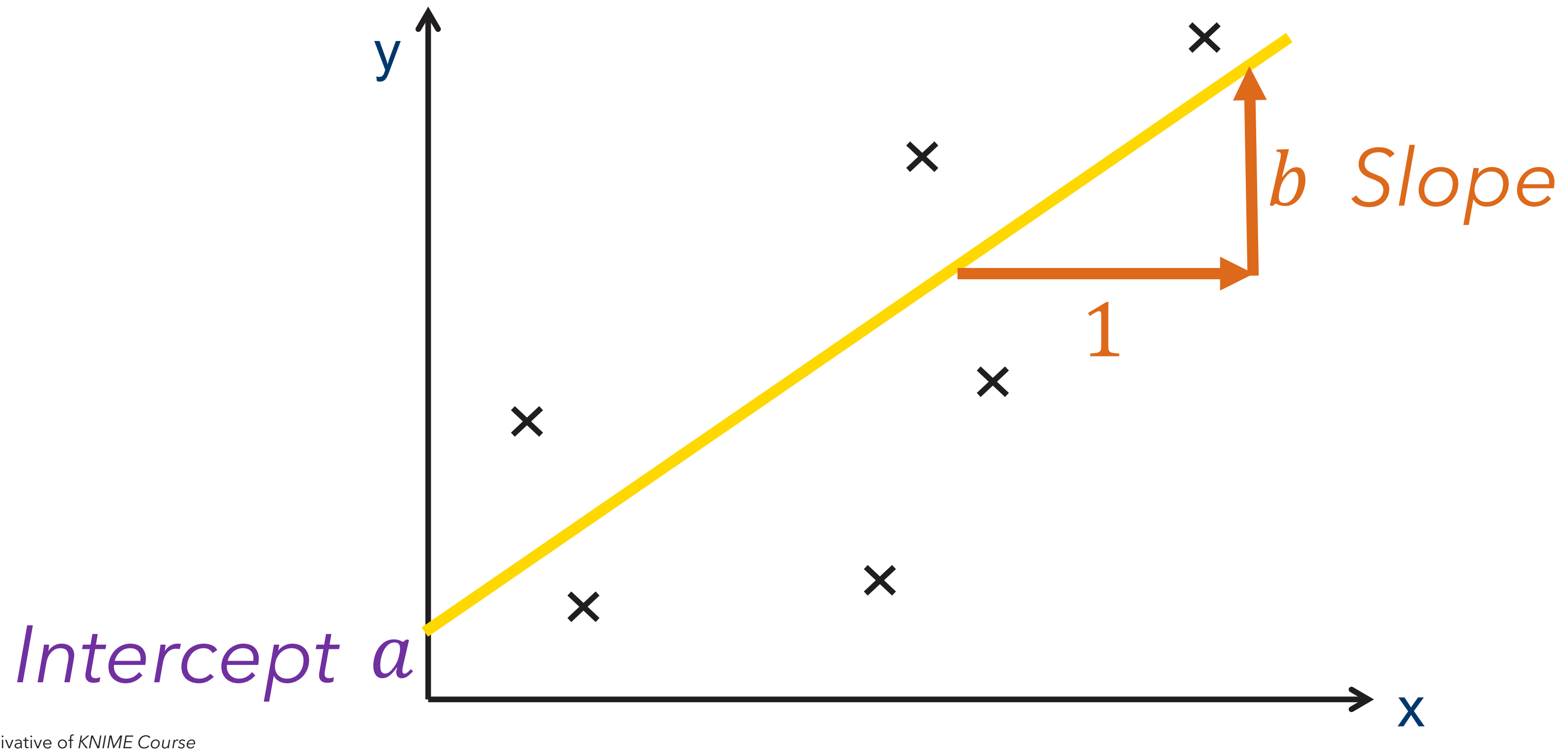
y = number of hours

Linear Regression: One Input Variable

- Given a data set with two continuous attributes, x and y
- There is an approximate linear dependency between x and y

Intercept Slope

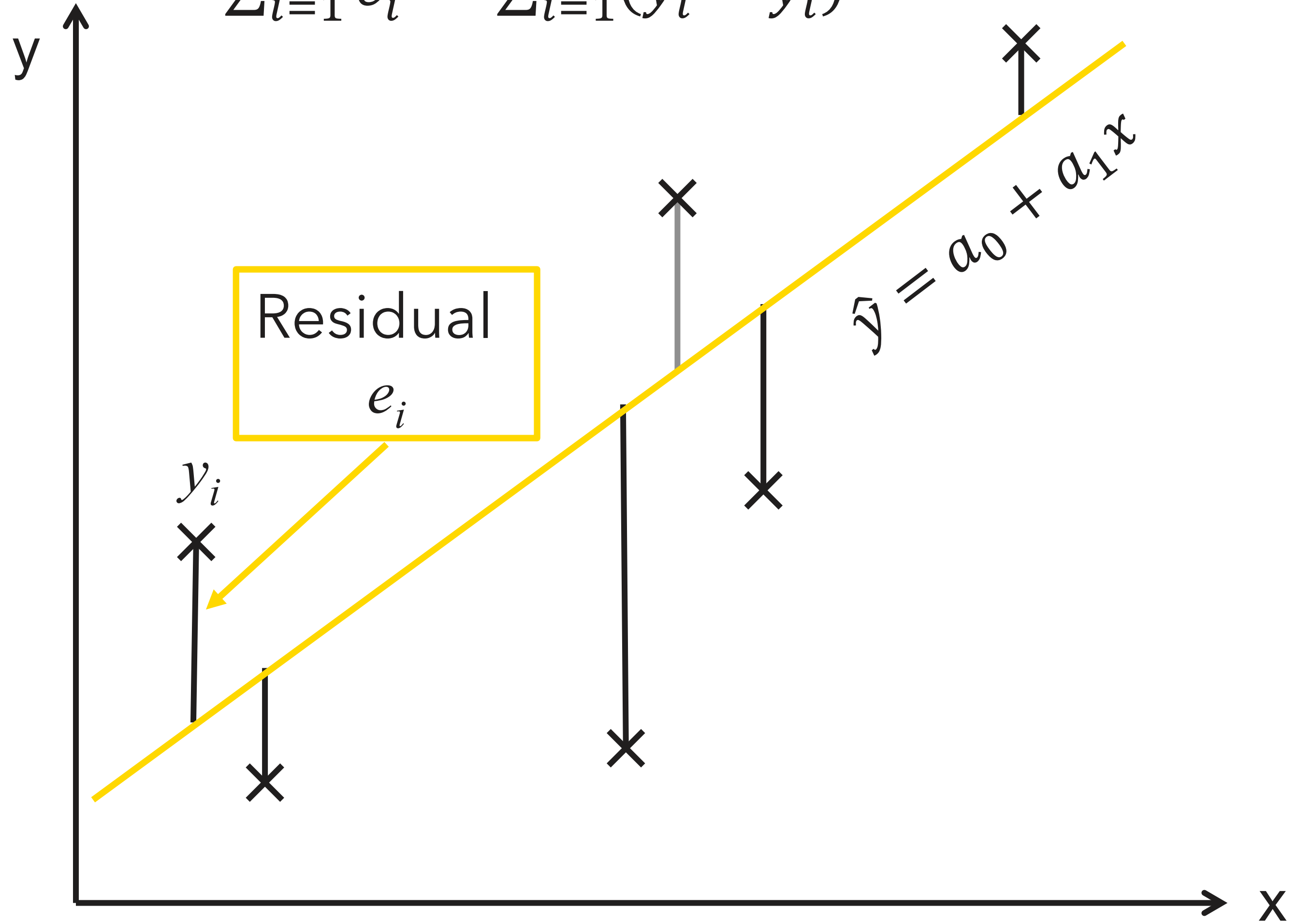
$$y \approx \boxed{a} + \boxed{b}x$$



Simple Linear Regression

Optimization goal: minimize sum of squared residuals

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$



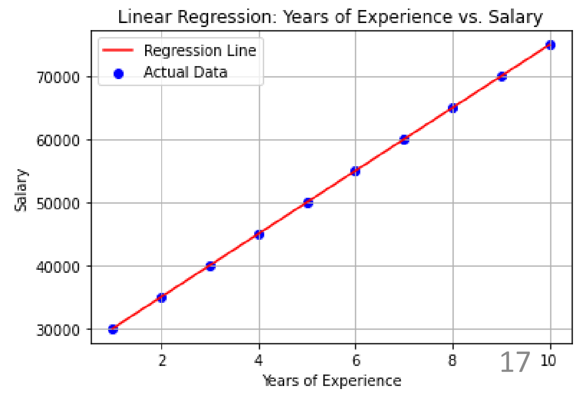
Linear Regression in Python (1/2)

```
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# 1. Prepare the data
# Sample independent variable (e.g., years of experience)
X = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]).reshape(-1, 1)
# Reshape to a 2D array as scikit-learn expects feature data in this format

# Sample dependent variable (e.g., salary)
y = np.array([30000, 35000, 40000, 45000, 50000, 55000, 60000, 65000, 70000, 75000])

# 2. Create and train the model
model = LinearRegression() # Create a Linear Regression model instance
model.fit(X, y) # Train the model using the independent (X) and dependent (y)
variables

# 3. Get model parameters
slope = model.coef_[0] # The coefficient (slope) of the regression line
intercept = model.intercept_ # The intercept of the regression line
print(f"Slope (coefficient): {slope:.2f}")
print(f"Intercept: {intercept:.2f}")
```

```
Slope (coefficient): 5000.00
Intercept: 25000.00
```

Linear Regression in Python (2/2)

```
# 4. Make predictions
y_pred = model.predict(X) # Predict y values based on the trained
model and X

# 5. Visualize the results
plt.scatter(X, y, color='blue', label='Actual Data’)
plt.plot(X, y_pred, color='red', label='Regression Line’)
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.title("Linear Regression: Years of Experience vs. Salary")
plt.legend()
plt.grid(True)
plt.show()
```



Simple vs. Multiple Regression

Predicts the values of the target variable *y*
based on a linear combination of
the values of the input feature(s) x_j

Two input features: $\hat{y} = a_0 + a_1 x_1 + a_2 x_2$

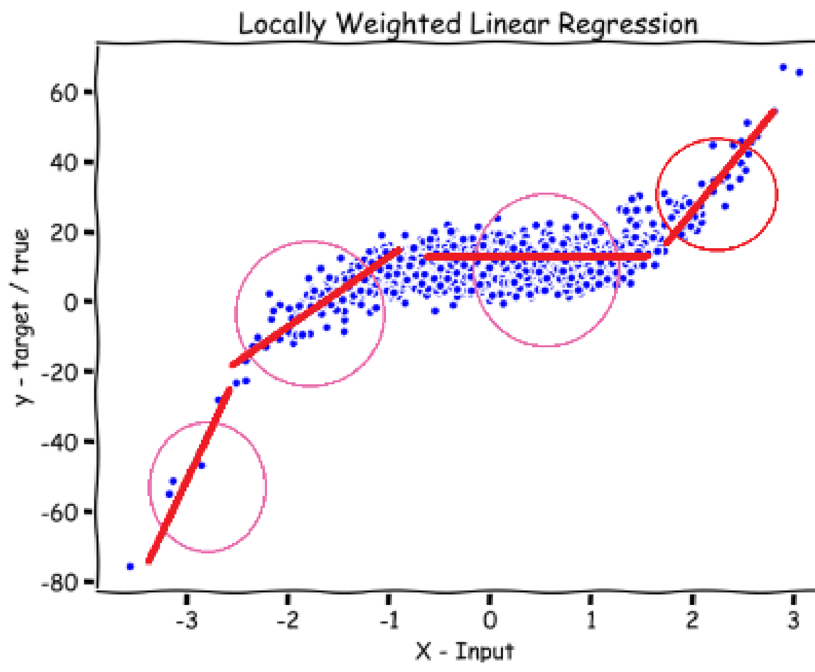
p input features: $\hat{y} = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_p x_p$

- Simple regression: one input feature → regression line
- Multiple regression: several input features → regression hyper-plane
- Residuals: differences between observed and predicted values (errors)
  Use the residuals to measure the model fit

Linear Regression: Summary

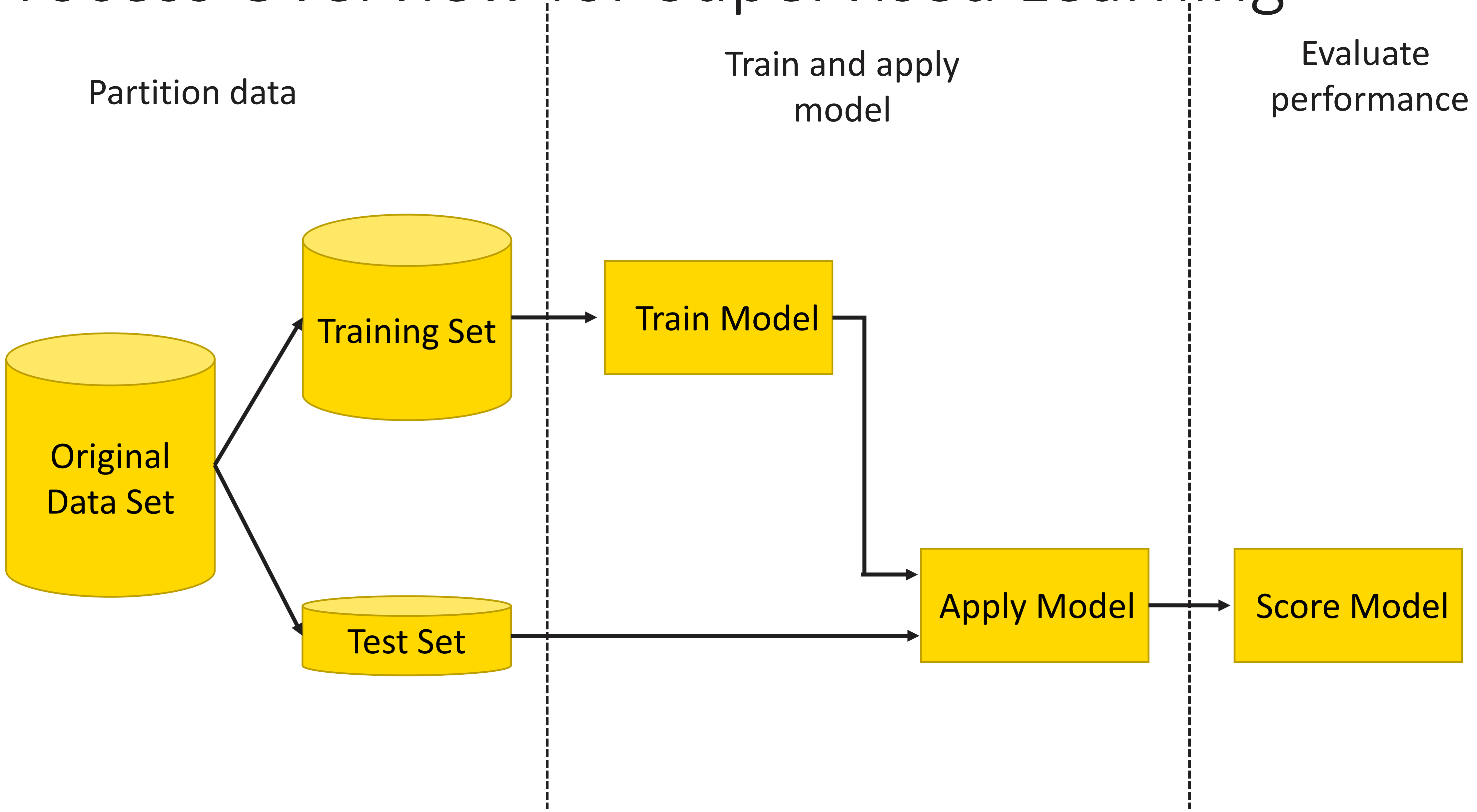
- Positive:
  - Strong mathematical foundation
  - Simple to calculate and to understand (For moderate number of dimensions)
  - High predictive accuracy (In many applications)

- Negative:
  - Many dependencies are non-linear (Can be generalized)
  - Model is global and cannot adapt well to locally different data distributions But see locally weighted regression



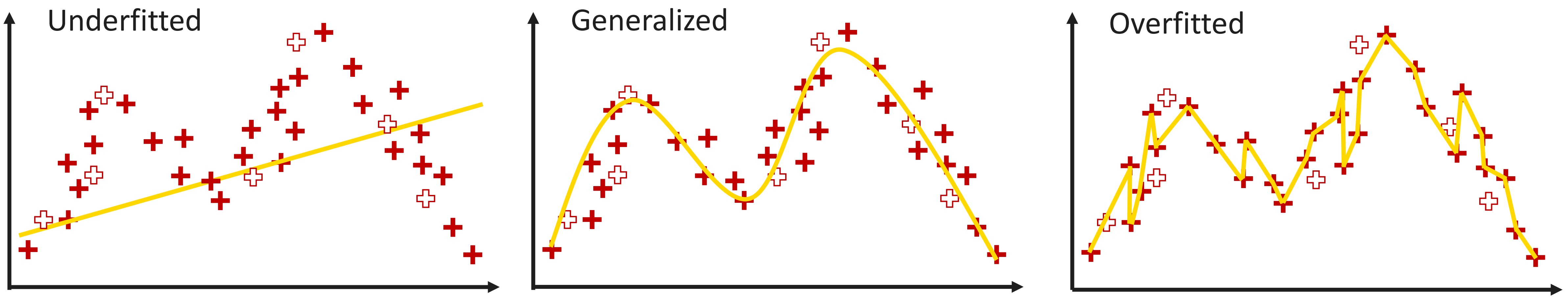
https://towardsdatascience.com/locally-weighted-linear-regression-in-python-3d324108efbf

Process Overview for Supervised Learning



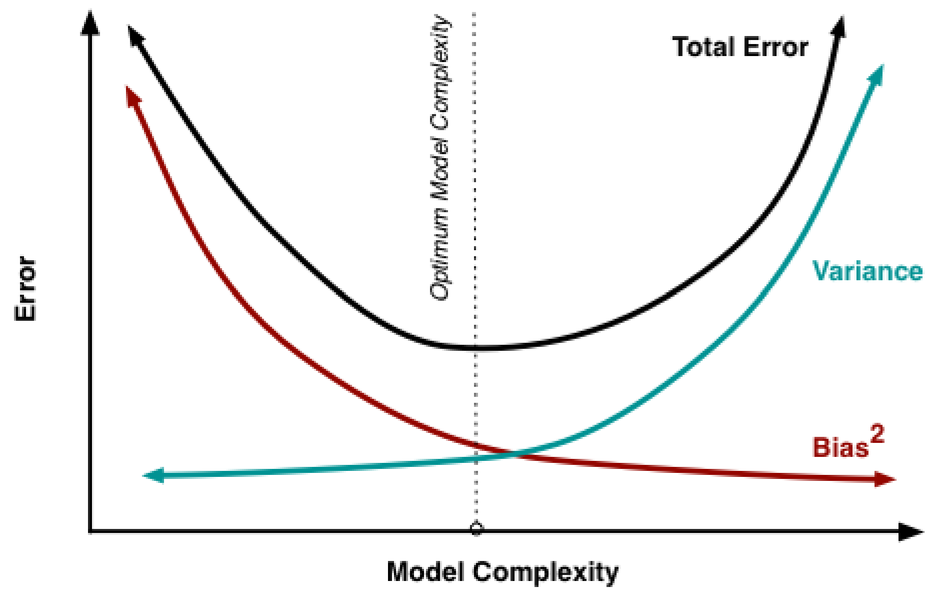
Overfitting vs Underfitting

Underfitting	**Overfitting**
▪ A model that can neither model the training data nor generalize to new data	▪ Model that fits the training data too well, including details and noise ▪ Negative impact on the model's ability to generalize



Bias vs. Variance Tradeoff

- **Bias**: Measures expected deviation from true value of function – training error
  - **Doesn't fit the training data** perhaps because the model is of the wrong form (e.g. model is linear when data is non-linear)
- **Variance**: Measures deviation from expected value based on data sampling – test error
  - **Fits the training data but not the test data**, perhaps because test data is very different from training set, or training set is too small



Kyoosik Kim, "Ridge Regression for Better Usage," Jan 2, 2019, https://towardsdatascience.com/ridge-regression-for-better-usage-2f19b3a202db

Regularization to Reduce Test Error

- By adding many more variables, we can model all the data (by effectively memorizing all inputs) and minimize bias, but this increases variance and doesn’t help the model generalize to non-training instances
- To counteract this, we use regularization – a modification that helps improve generalization accuracy without affecting training accuracy
- Regularization terms in regression impose a penalty for including too many terms that would otherwise overfit the data

Ridge Regression

- Ridge regression is a special type of regression where the ridge parameter helps stabilize model coefficients when predictors are in a condition of near-collinearity; gives a bias to important features
  - Loss function is sum of squared errors plus λ * sum of squared coefficients

$$L = \sum (\hat{Y}_i - Y_i)^2 + \lambda \sum \beta^2$$

- Benefits
  - Ability to handle a large number of predictors regardless of feature dependencies
  - Higher accuracy compared to many other model types
  - Faster to build even for a hundred predictors and millions of data points

Kyoosik Kim, ”Ridge Regression for Better Usage,” Jan 2, 2019,
https://towardsdatascience.com/ridge-regression-for-better-usage-2f19b3a202db

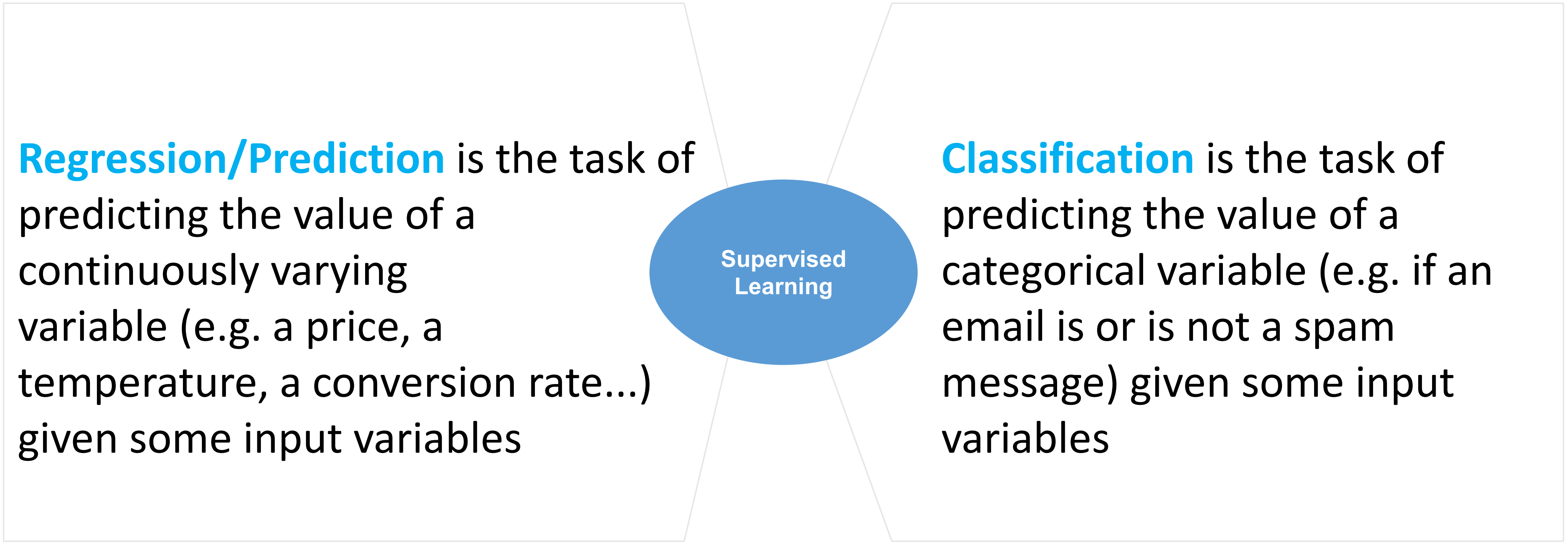
Lasso Regression

- Lasso regression is similar to ridge regression but drives weights of irrelevant variables to 0

$$L = \sum(\hat{Y}_i - Y_i)^2 + \lambda \sum |\beta|$$

- The sum of the absolute values of the coefficients is called the **L$_1$-norm**
  - In contrast, the sum of the squared coefficients used in ridge regression is called the **L$_2$-norm**

Classification vs. Regression/Prediction



Regression/Prediction is the task of predicting the value of a continuously varying variable (e.g. a price, a temperature, a conversion rate...) given some input variables

Classification is the task of predicting the value of a categorical variable (e.g. if an email is or is not a spam message) given some input variables

Logistic Regression as Classification

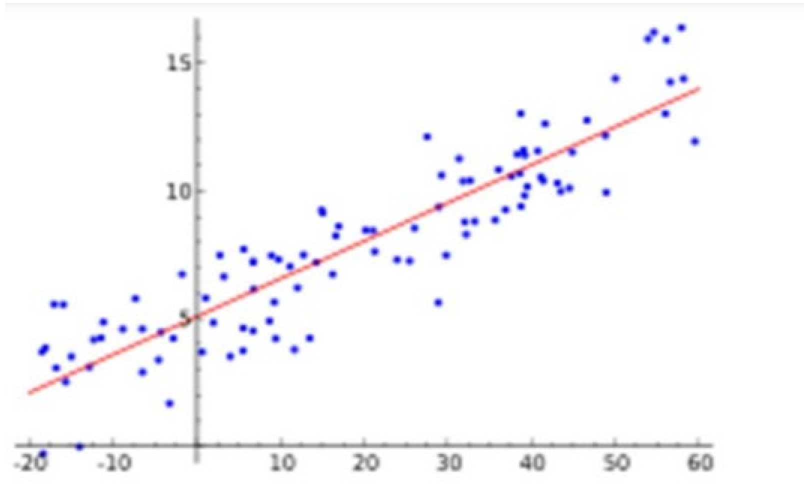
- Classification: Which of a set of categories (sub-populations) does an observation belong to?
  - A given email into "spam" or "non-spam" classes.
  - A customer will "purchase" or "not purchase"
- Binary Logistic Regression, or **logit** regression is used to predict a binary response based on one or more predictor (or independent) variables (features), The logit model solves these problems:

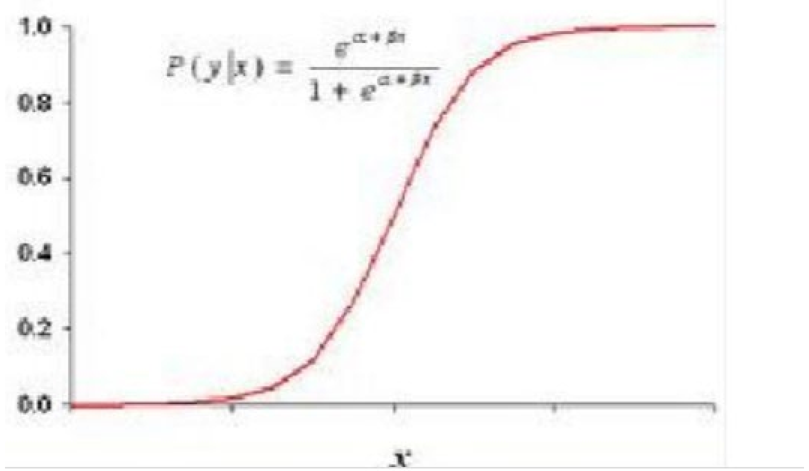
$$\ln[p/(1-p)] = \beta_0 + \beta_1 X + e$$

- Where
  - p is the probability that the event Y occurs, p(Y=1)
  - p/(1-p) is the "odds ratio", and
  - ln[p/(1-p)] is the log odds ratio, or logit

Linear vs. Logistic Regression

- In linear regression, we are predicting a continuous quantity Y, which varies in proportion to X
- In logistic regression, we are predicting the likelihood that Y=1 (rather than 0) given certain values of Xs.
- If X and Y have a positive linear relationship it indicates the probability that Y=1 will increase as value of X increases

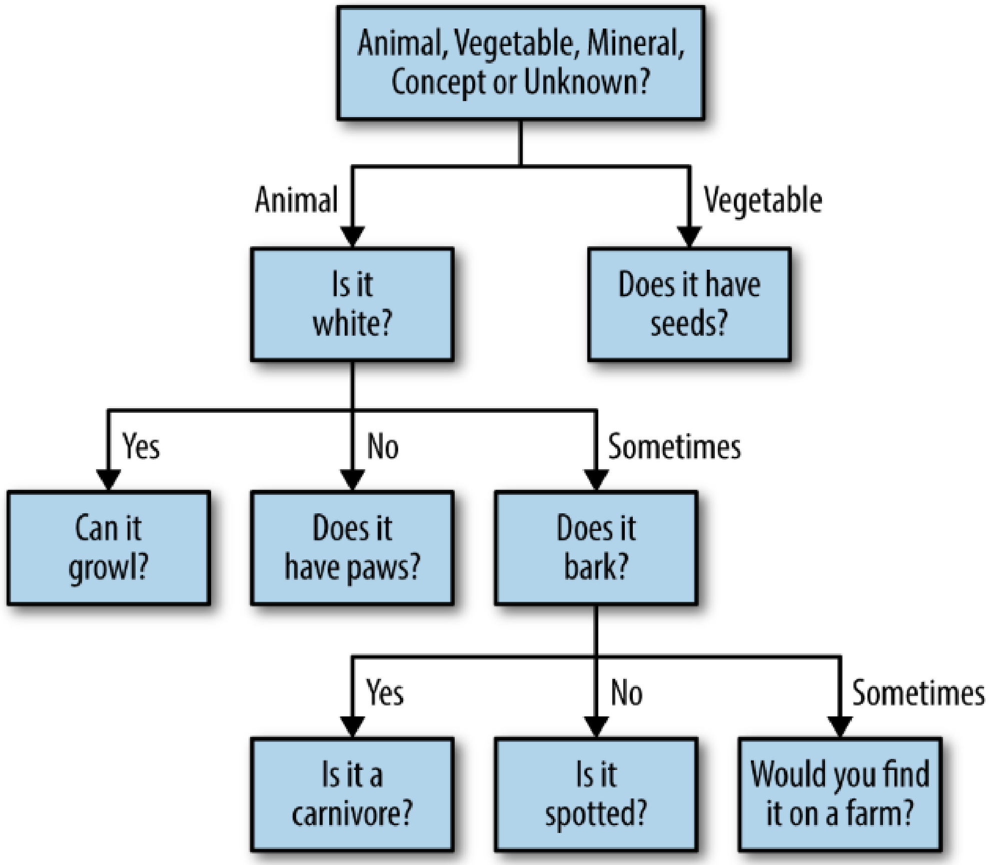




P. Wu, et. al Comparison of linear and logistic regression for segmentation , 2014
https://www.casact.org/sites/default/files/presentation/rpm_2014_handouts_paper_2693_handout_1986_0.pdf

Decision Trees

- A **decision tree** uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes.
- Decision tree as a predictive model maps observations about an item to conclusions about the item's target value.



Hebron, P. Machine Learning for Designers, https://www.oreilly.com/library/view/machine-learning-for/9781491971444/#toc-start

Building Decision Trees

- Tree is grown from the root node, with each iteration splitting the data into two further sub-segments
- The algorithm stops when threshold conditions are met (e.g., exposure within the node becomes too small or maximum depth of tree has been reached)
- Pruning removes branches that make use of features having low importance (as judged by impact on accuracy)

Decision Tree in Python using Scikit-learn (1/3)

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Sample Data (replace with your actual data)
data = {
    'Feature1': [10, 20, 15, 25, 30, 12, 22, 18],
    'Feature2': ['A', 'B', 'A', 'C', 'B', 'A', 'C', 'B'],
    'Target': [0, 1, 0, 1, 1, 0, 1, 0]
}
df = pd.DataFrame(data)
# Encode categorical features
df['Feature2'] = df['Feature2'].astype('category').cat.codes
X = df[['Feature1', 'Feature2']]
y = df['Target']
```

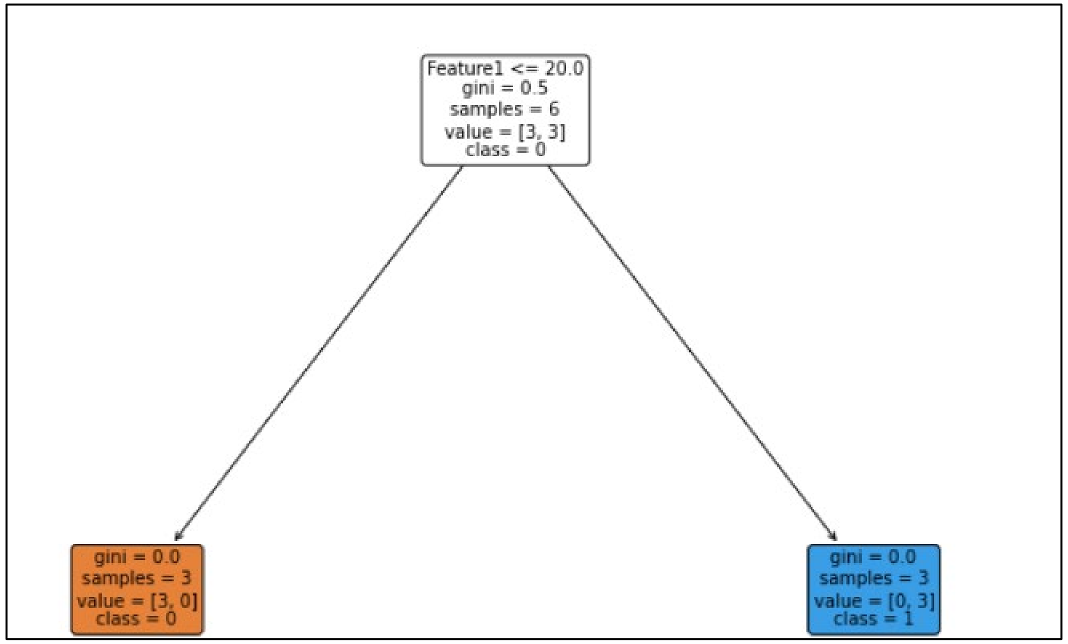
Decision Tree in Python using Scikit-learn (2/3)

```python
# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train the classifier
dt_classifier = DecisionTreeClassifier(max_depth=3, random_state=42)
dt_classifier.fit(X_train, y_train)

# Make predictions
y_pred = dt_classifier.predict(X_test)

# Evaluate
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.5
```

Decision Tree in Python using Scikit-learn (3/3)

```
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree
feature_names = ['Feature1', 'Feature2_encoded']
class_names = [str(name) for name in dt_classifier.classes_]
# Create a figure and plot the tree
plt.figure(figsize=(15, 10))
plot_tree(
    dt_classifier,
    feature_names=feature_names,
    class_names=class_names,
    filled=True,
    rounded=True,
    fontsize=10
)
plt.show()
```



Determining the Best Split

- Choose the attribute that yields the highest **information gain** in dividing the remaining observations
    - Yielding subsets of observations separated by outcome variable value that are least likely to be derived by chance
- Information gain defined in terms of **entropy** (uncertainty of a random variable measured in bits). For *n* outcomes, each with likelihood p_i

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

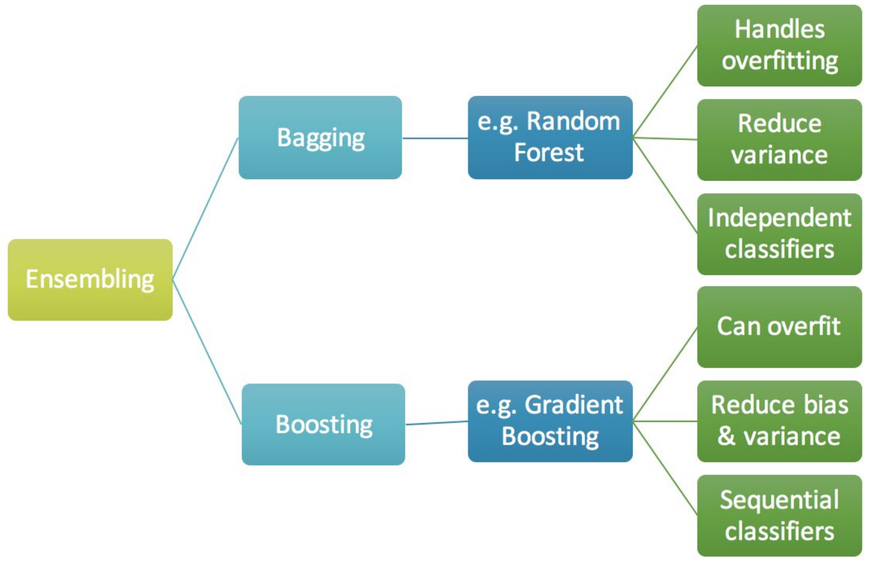
- Entropy is
    - 1 bit for a coin flip (two equally likely outcomes)
    - 2 bits for four-sided die (four equally likely outcomes)
    - .08 bits for a loaded coin that yields heads 99% of the time
- Split the decision using the attribute that yields the lowest weighted average entropy of the child nodes
- Other methods use **chi-square** and **Gini coefficient**

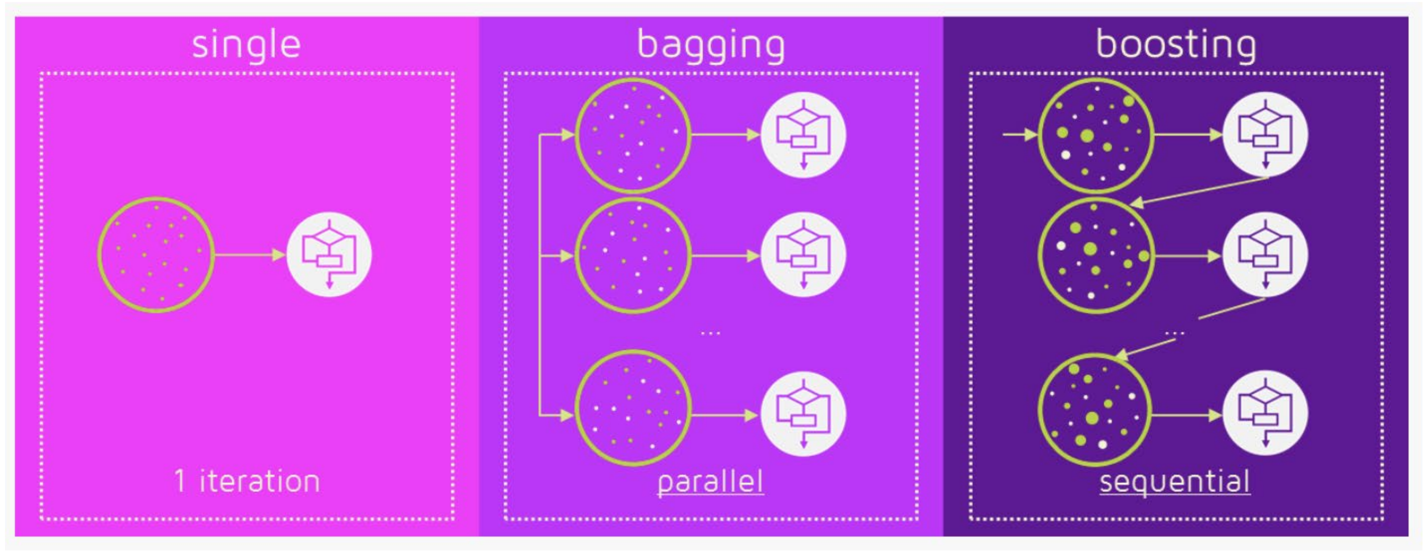
Decision Trees: When to Use Them?

- When many types of input variables are present: Binary, categorical, ordinal, continuous
- When large data sets are available to cover the space
- Where relationships are not linear

Ensemble Methods

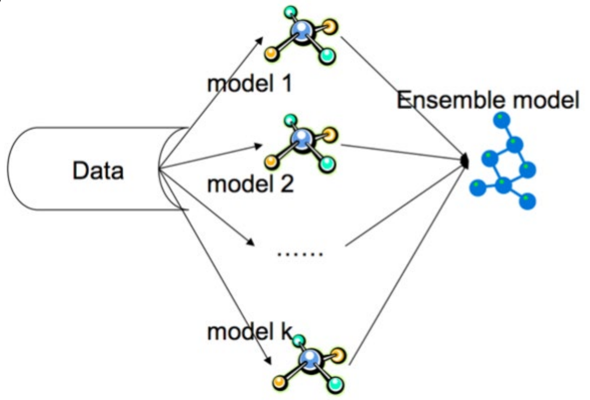
- Uses multiple models at a time to combine strengths of each
- Can reduce bias and overfitting, but may reduce interpretability
- Two ways of “ensembling”: Bagging and Boosting





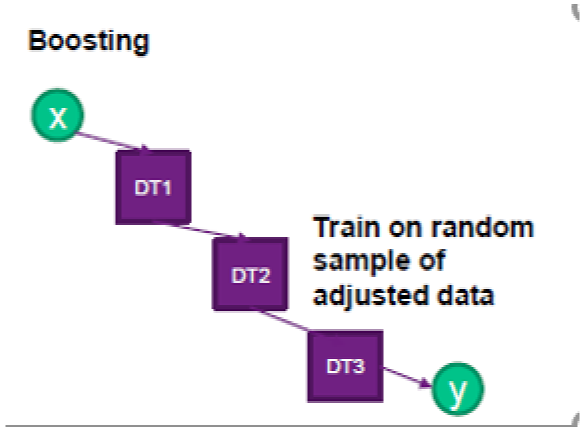
Bagged Decision Trees and Random Forests

- Decision trees are sensitive to order of training examples and irrelevant factors
- Multiple decision trees can be combined in an **ensemble** using **bagging** (short for bootstrap aggregation)
  - Prediction from each tree averaged together, weighted by tree accuracy
- **Random forests** are bagged decision trees trained by
  - Random subsets of data, to reduce variance
  - Random subsets of features, to prune irrelevant factors



Other Supervised Learning Models

- **Support vector machines**: Find hyperplane separated points in high-dimensional space
- **Boosting**:
  - Each model trained on the results of previous model
  - More weight is given in subsequent models to observations that were not classified correctly previously
  - **Gradient boosting** adds estimator that fits to the residual error of previous model
  - Example: **Gradient-tree boosting** works on decision trees.



Conclusion

- AI problems can come from any task that requires intelligence, but it is useful to take both a functional (horizontal) and an industry/mission (vertical) view
- Machine learning techniques generalize from observed data
  - Supervised learning predicts quantities or classes from labeled examples
  - Unsupervised learning finds interesting data groupings
- Supervised learning techniques include
  - Linear regression
  - Logistic regression
  - Decision trees
  - Others: Random forests, Gradient-boosted trees, Support vector machines
- **Next class** (Sep 4): Unsupervised Methods

Conclusion

- AI’s 80-year history is almost as long as computers have been around, but AI has been especially active in the last decade or two due to faster computers, more data and deep learning
- AI problems can come from any task that requires intelligence, but it is useful to take both a functional (horizontal) and an industry/mission (vertical) view
- **Next class (Sep 2):** Unsupervised Machine Learning
    - Read over HW1 along with the readings; get started early!
    - Be prepared to present an AI application idea in class (2 mins)
        - What problem are you solving?
        - What would be the impact?