Supervised Learning

TACC Machine Learning Institute

August 3, 2021

PRESENTED BY:

Kelly Pierce

Research Associate

Scalable Computational Intelligence

Schedule

Morning Session, 9:30 - 11:00a

- Supervised Learning Overview
- Classification methods: KNN, SVM, Decision Trees

Afternoon Session, 2:00 - 3:30

- Regression methods: linear, logistic, non-linear

What I assume you know...

- Bash experience (file system navigation, basic commands)
- Python experience (variable assignment, basic data types)
- Data experience (basic statistics, descriptive visualization)

Reach out on Zoom chat with questions!

What is machine learning?

Definition from T. Mitchell (1997). *Machine Learning book*:

> “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at the tasks improves with the experiences.”
>
> --- (Mitchell 1997)

- Learn from past experiences
- Improves with experience

Why use machine learning?

Existing models are insufficient for the problem

- too many unknown variables
- many factors
- missing data

Expertise and experience are hard to explain -- a human can make the right decision, but cannot explain how

Outcome changes over time

Transfer knowledge from one case to another

Goal of machine learning

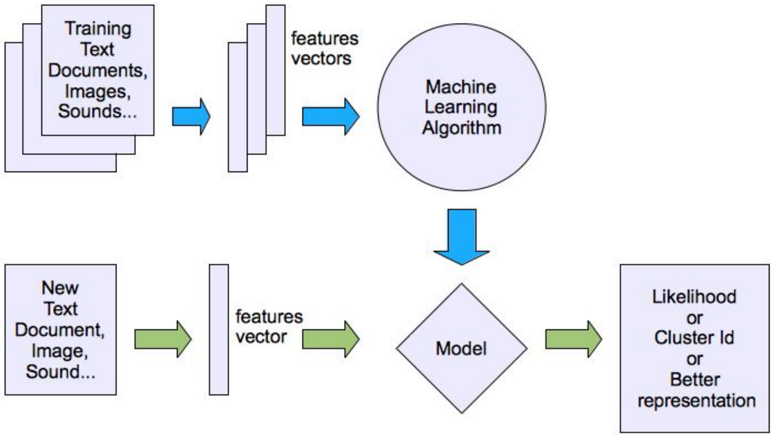
Learn a general model from existing data

- Useful when data are cheap, abundant
- Helps tease out noise and identify hidden variables

Identify model or structural pattern that is a useful, **general** approximation to the data goal -- not just descriptive of a single dataset.

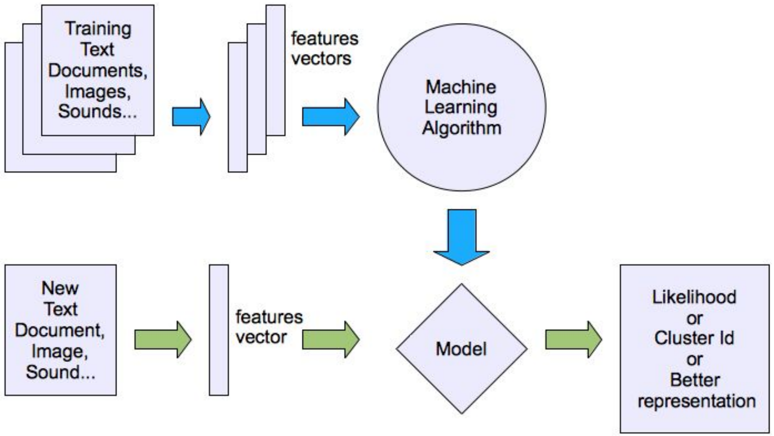
Machine Learning Taxonomy

<u>Unsupervised learning</u>
learn data structure

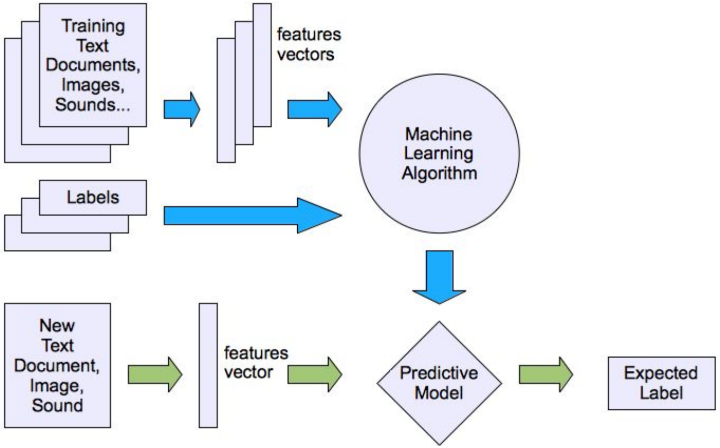


Machine Learning Taxonomy

<u>Unsupervised learning</u>
learn data structure



<u>Supervised learning</u>
learn data model



High-performance computing empowers machine learning

Mass storage

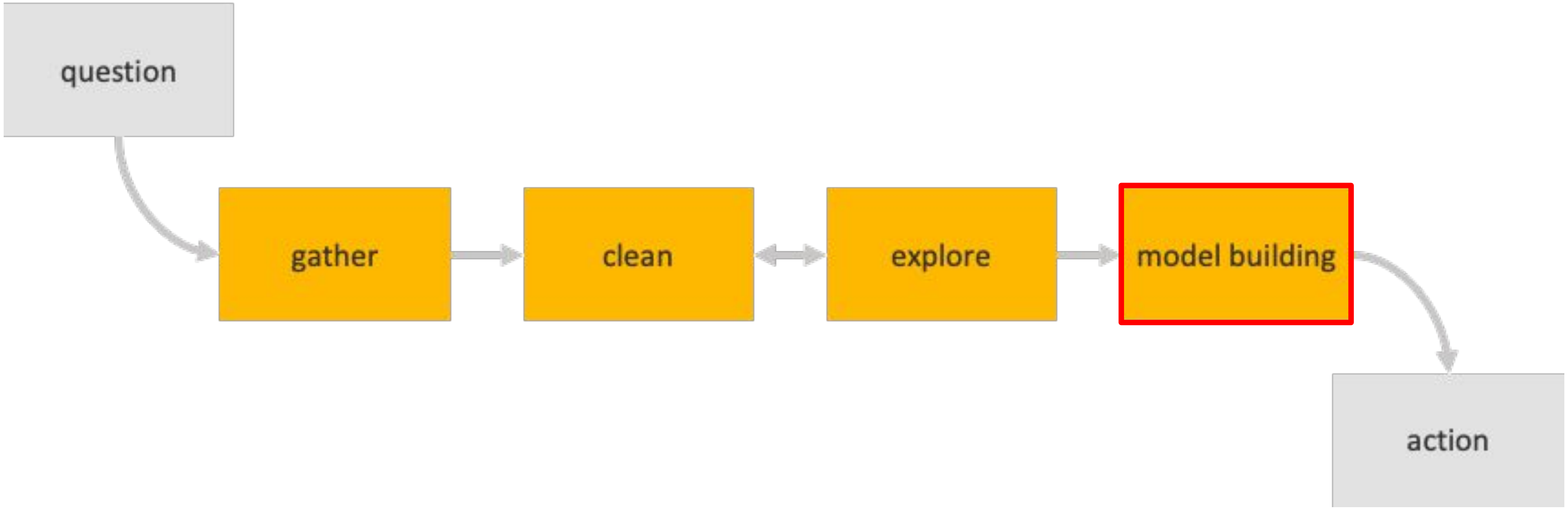
- more data
- more past “experience”

Faster computations

- more and faster memory
- make complex solutions practical through scaling

TACC

Building models is a very small part of the machine learning workflow



Supervised Learning

Statistical models for supervised learning

Classification

- K Nearest Neighbors (KNN)
- Support vector machine
- Decision trees
- Random forest
- Logistic regression

categorical data

Regression

- Linear regression
- Non-linear regression
- Bayesian regression
- Random forest regression

numeric data

Training and testing

Training

- The process of making a system to learn a model
- Data to be “observed” by the learning system

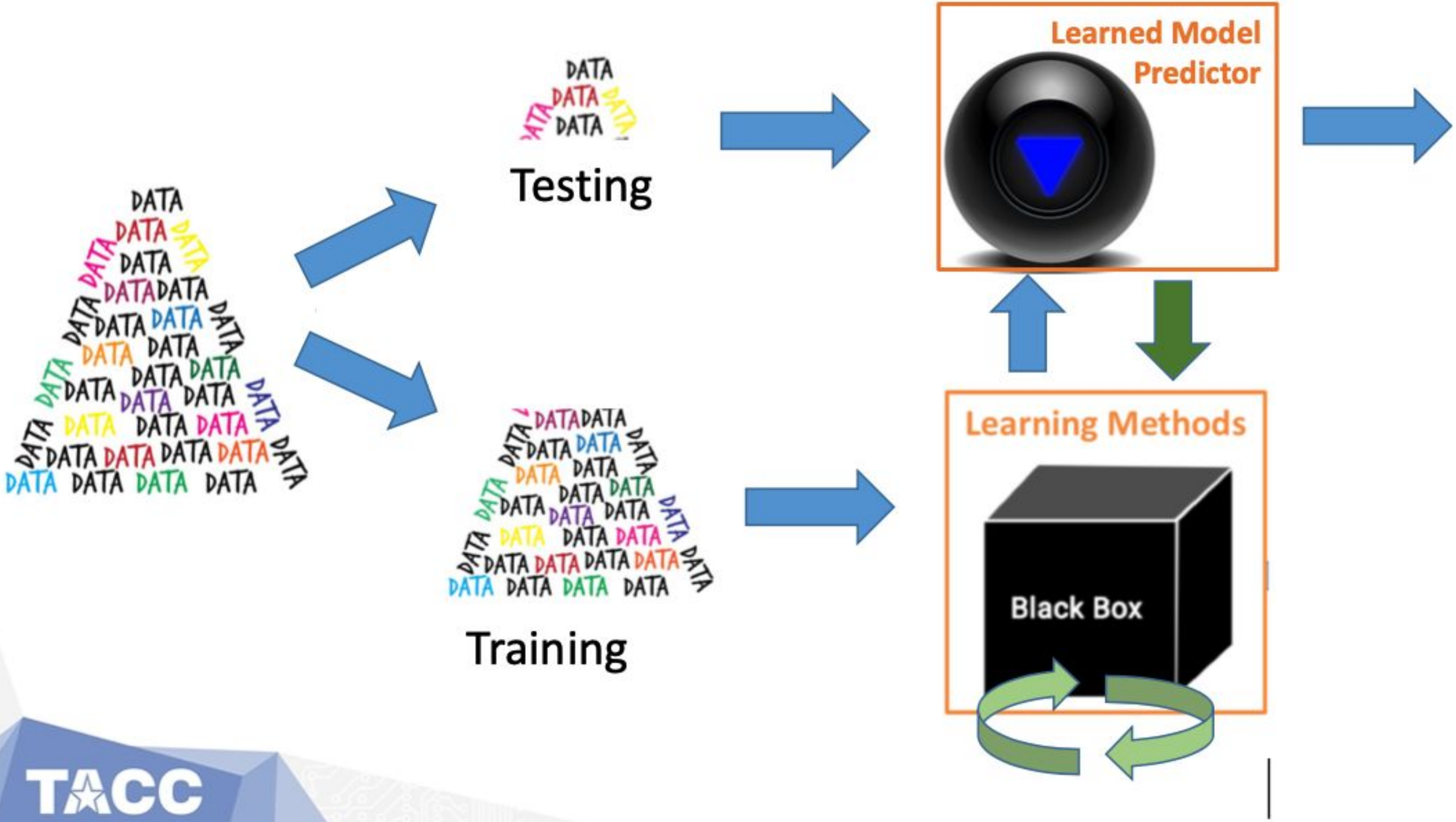
Testing

- The process of evaluating model performance
- Data are not observed by the learning system

Prediction

- Data are not used in training or testing

Many machine learning workflows use 80% of available data to train, and 20% of available data to test (**80/20 split**)


Testing
Learned Model
Predictor
Training
Learning Methods
Black Box

TACC

Performance metrics for supervised learning

Classification

- Precision
- Recall
- Accuracy

categorical data

Regression

Cost or loss function:

- Squared error
- Root mean squared error

numeric data

Supervised Learning: Classification Methods

K Nearest Neighbor (KNN) Classification

- If two data objects are similar (close in value), they are likely from the same class
- Workflow
  - Start with a set of observations with class labels
  - Define a distance (similarity) measure
  - Retrieve the k nearest neighbors for each observation
  - Assign the class label based on the most common label of the neighbors
  - Evaluate performance (how often was the predicted label correct?)

TACC

KNN Example: tax evasion detection

Features / Class labels

Refund	Marital Status	Taxable Income	Evade
yes	single	125k	no
no	married	100k	no
no	single	70k	no
yes	married	120k	no
no	divorced	95k	yes
no	married	60k	no
yes	divorced	220k	no
no	single	85k	yes
no	married	75k	no
no	single	90k	yes

Do you think this record is associated with tax evasion?
- Consider k=1 nearest neighbors
- Consider k=2 nearest neighbors
- Consider k=3 nearest neighbors

Refund	Marital Status	Taxable Income	Evade
no	single	85k	???

KNN Example: tax evasion detection

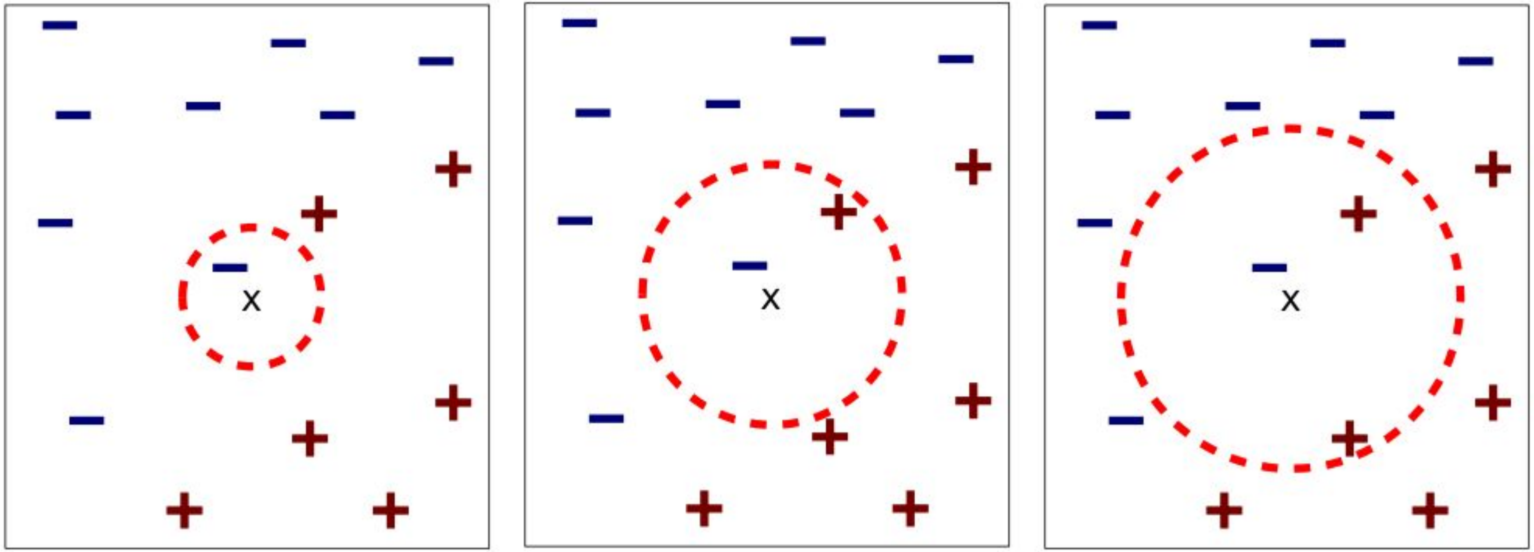
Features **Class labels**

Refund	Marital Status	Taxable Income	Evade
yes	single	125k	no
no	married	100k	no
no	single	70k	no
yes	married	120k	no
no	divorced	95k	yes
no	married	60k	no
yes	divorced	220k	no
no	single	85k	yes
no	married	75k	no
no	single	90k	yes

Do you think this record is associated with tax evasion?
- Consider k=1 nearest neighbors
- Consider k=2 nearest neighbors
- Consider k=3 nearest neighbors

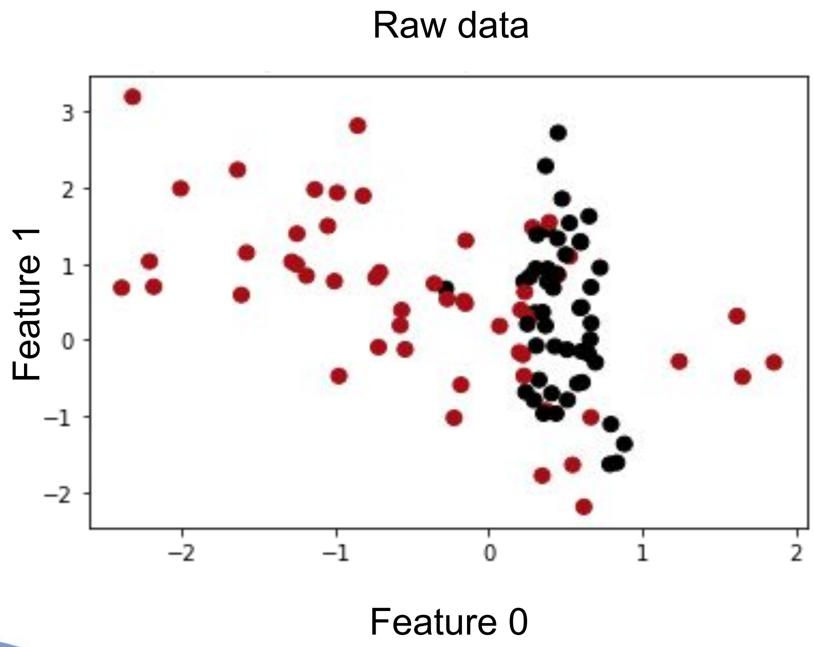
Refund	Marital Status	Taxable Income	Evade
no	single	85k	???

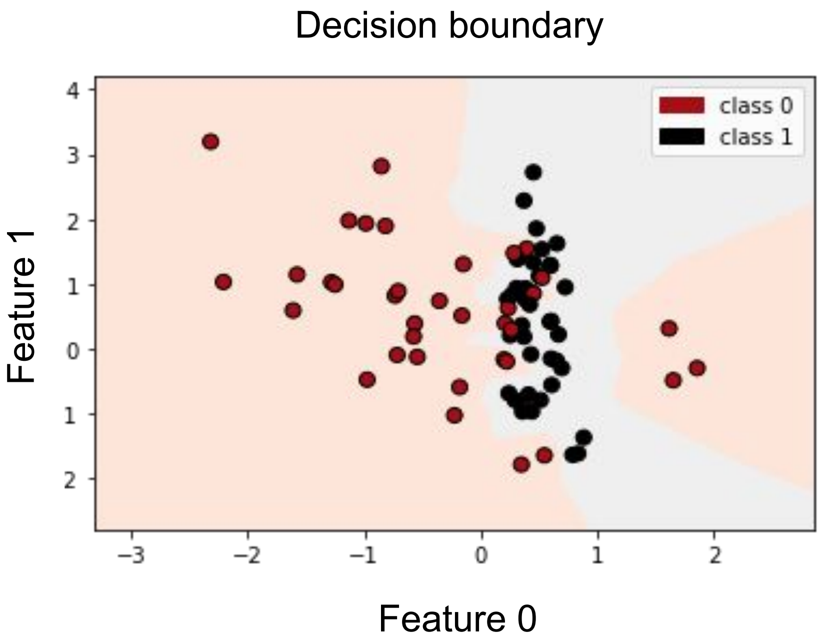
KNN Classification



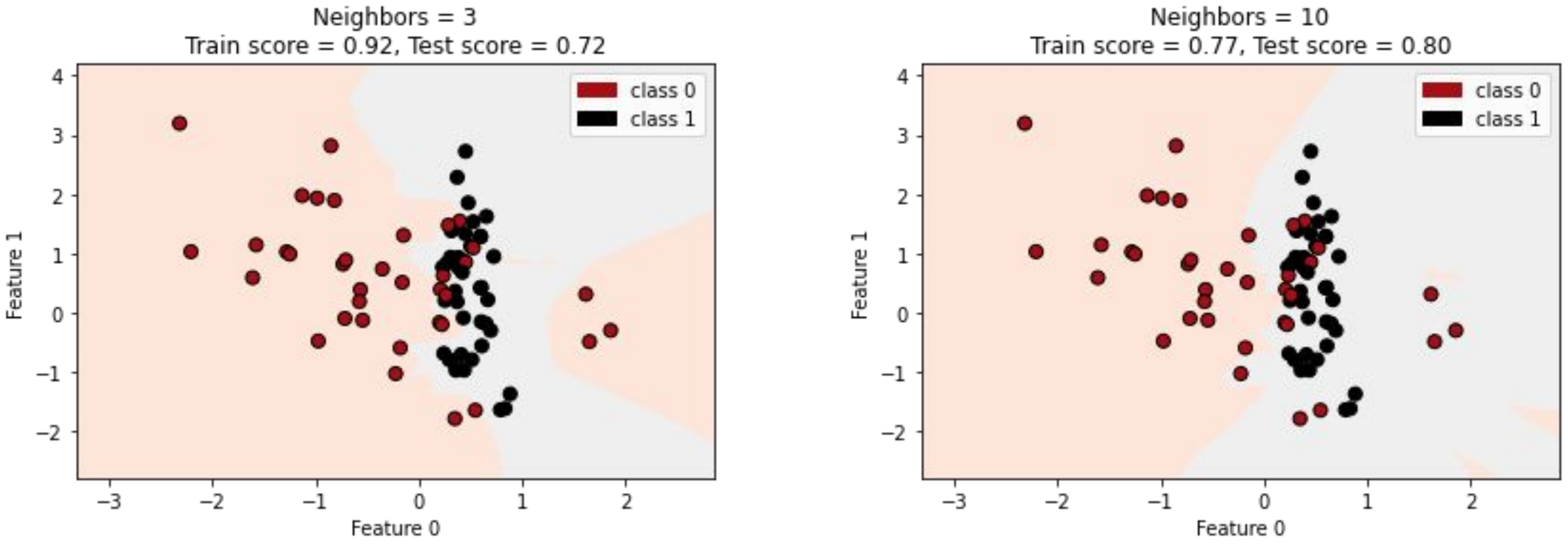
(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Example binary decision boundary for KNN





Choice of K impacts decision boundary



What happens as K is increased?

Binary classification performance

Confusion matrix

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

Binary classification performance

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

$$\text{Recall} = \frac{a}{a+b} = \frac{TP}{TP+FN}$$

Correctly classified positive / all positive
Also known as “specificity”

$$\text{Precision} = \frac{a}{a+c} = \frac{TP}{TP+FP}$$

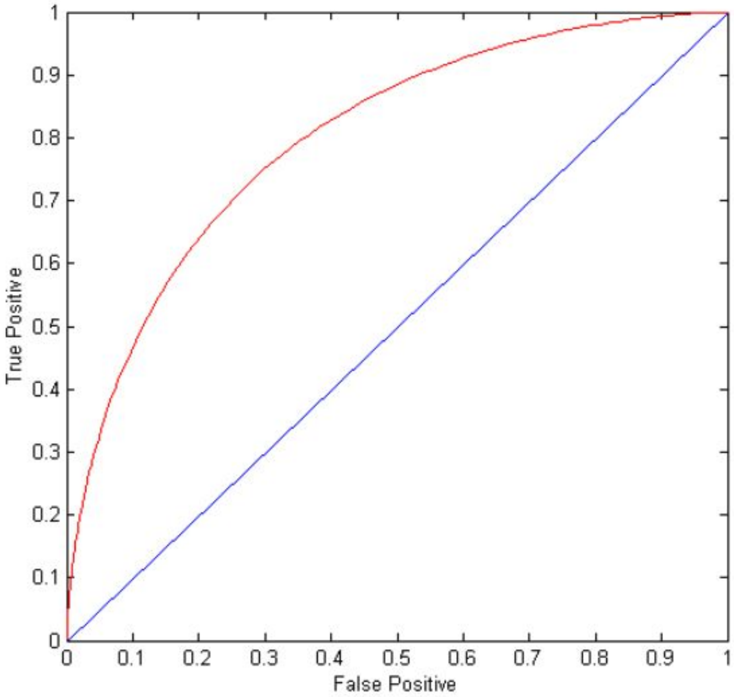
Correctly classified positive / all classified positive

Binary classification performance

Receiver-operating characteristic (ROC) curve

- (0, 0): classify everything as negative
- (1, 1): classify everything as positive
- (1, 0): classify all true positives with no false positives

Goal is to reach area under curve (AUC) close to 1



Hands-On Exercise

Jupyter Set-Up and KNN Classification

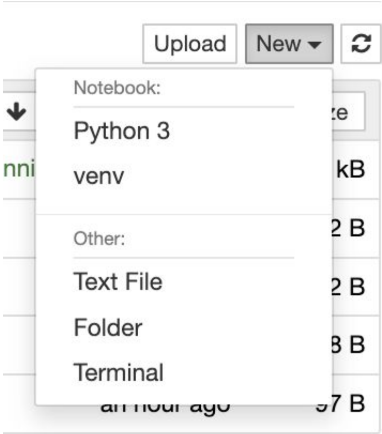
Start the Vis Portal Session

1. Log on to the Visualization Portal with your TACC **training account**.
2. Launch a Jupyter Notebook job from the Visualization Portal.
   a. Use reservation "**ML_Institute_day2**" on **Frontera**.
   b. Request a job time of **2 hours**

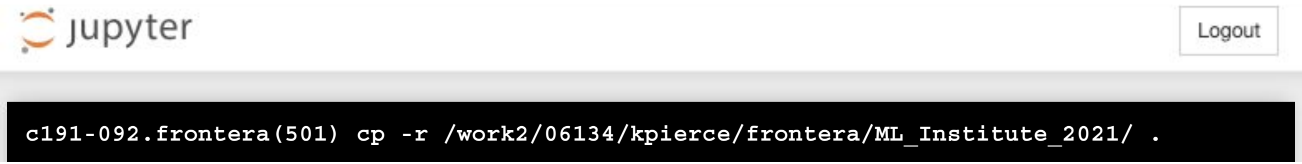
TACC

Copy the Course Materials to Current Dir

1. Open a new terminal session



2. Copy the materials



Here’s the command in larger font:

```
cp -r /work2/06134/kpierce/frontera/ML_Institute_2021/ .
```

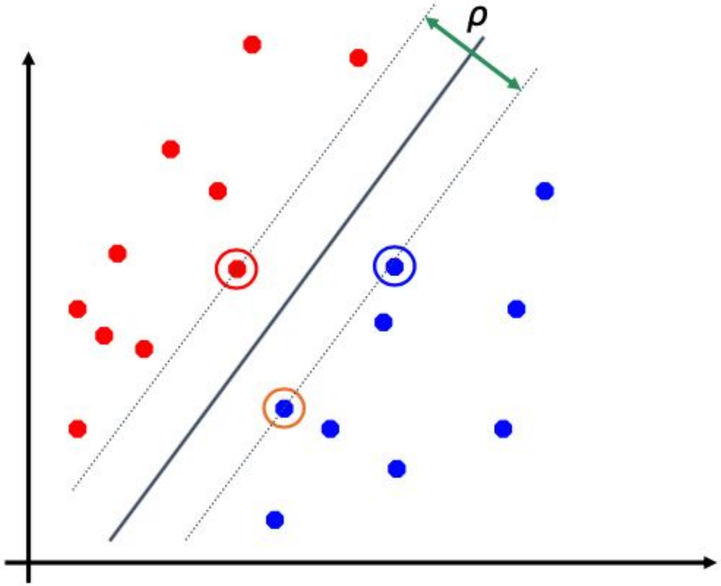
Launch the Jupyter Notebook

Navigate into the ML_Institute_2021 directory open the "SupervisedLearning.ipynb" notebook

Support vector machine (SVM)

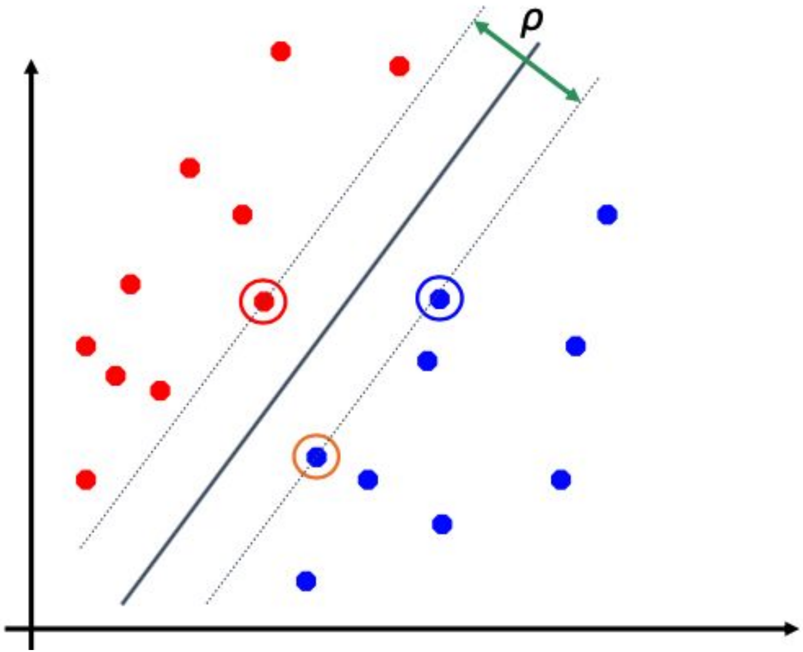
Generates optimal hyperplane through feature space to classify observations

- Points represent observations from two classes (**red**, **blue**).
- The line is a linear separator between the classes.
- The closest observations to the hyperplane define the support vectors.



Binary classification with SVM

- Find a line passing as far as possible fro both points
- The optimal separating hyperplane *maximizes* the margin of the training data.
- A line is bad if it passes too close to some points (noise sensitive)



Estimating a linear SVM

Hyperplane

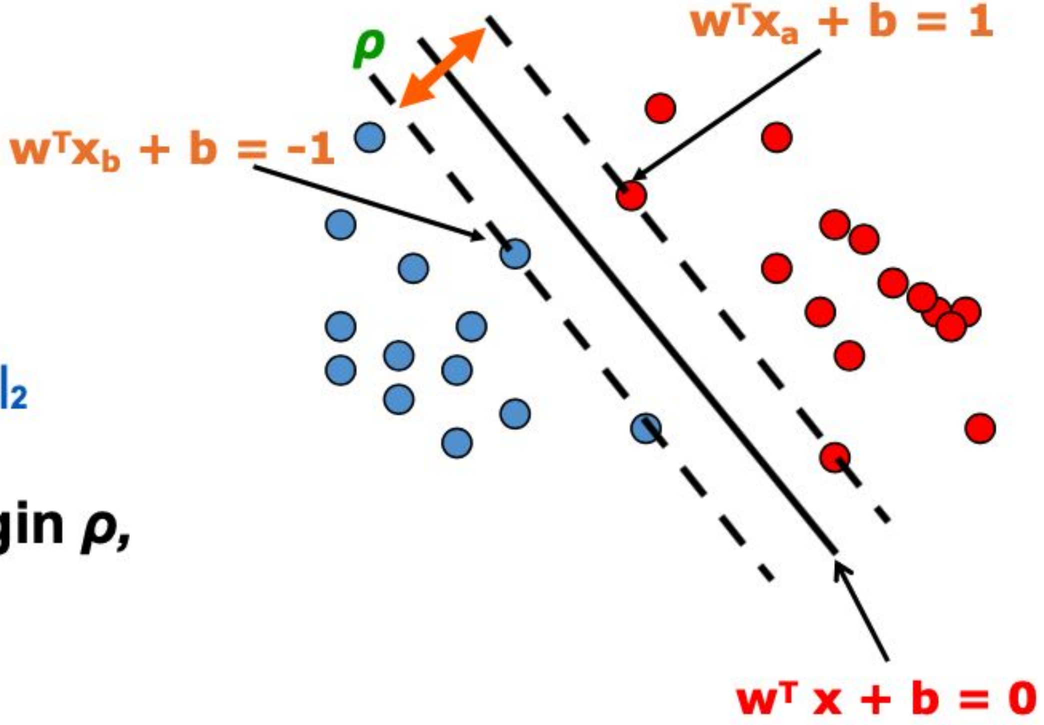
$$\mathbf{w}^T \mathbf{x} + b = 0$$

$$w^T(x_a - x_b) = 2$$

$$\rho = ||x_a - x_b||_2 = \mathbf{2/||w||_2}$$

Maximize the margin *ρ*,

***Minimize* ||w||**



Predicting with a linear SVM

Given a new point **x**, we can score its projection onto the hyperplane normal:

i.e., compute score: $\mathbf{w}^T\mathbf{x} + b$

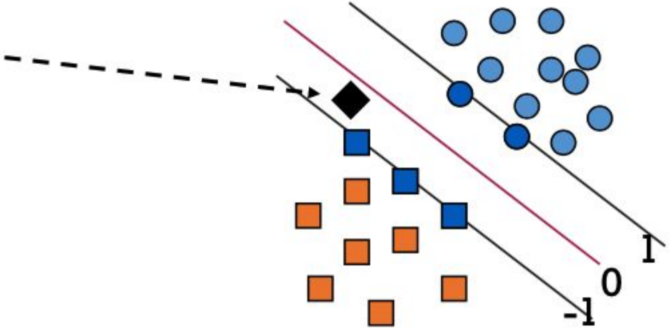
Decide class based on whether < or > 0

Can set confidence threshold *t*.

Score > *t*: yes

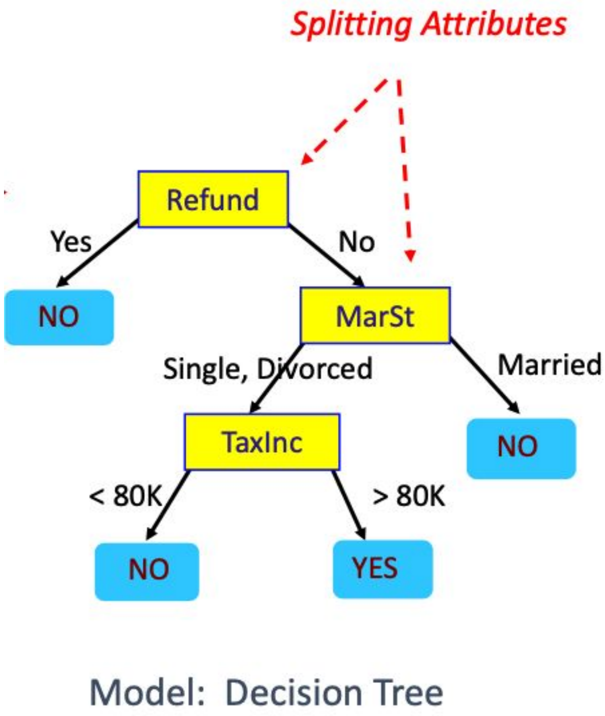
Score < -*t*: no

Else: don't know



Decision tree

Refund	Marital Status	Taxable Income	Evade
yes	single	125k	no
no	married	100k	no
no	single	70k	no
yes	married	120k	no
no	divorced	95k	yes
no	married	60k	no
yes	divorced	220k	no
no	single	85k	yes
no	married	75k	no
no	single	90k	yes



Select splits in decision trees using entropy

Maximize information gained in a split, while penalizing the number of small partitions.

Entropy at a given node t

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

Maximum ($\log n_c$) when records are equally distributed among all classes implying least information

Minimum (0.0) when all records belong to one class, implying most information

Information gain

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

n_i is number of records in partition i   p is the parent node

Measures reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)

Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

Gain ratio balances maximizing gain while penalizing many small partitions

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions

n_i is the number of records in partition i

Adjusts Information Gain by the entropy of the partitioning (SplitINFO).

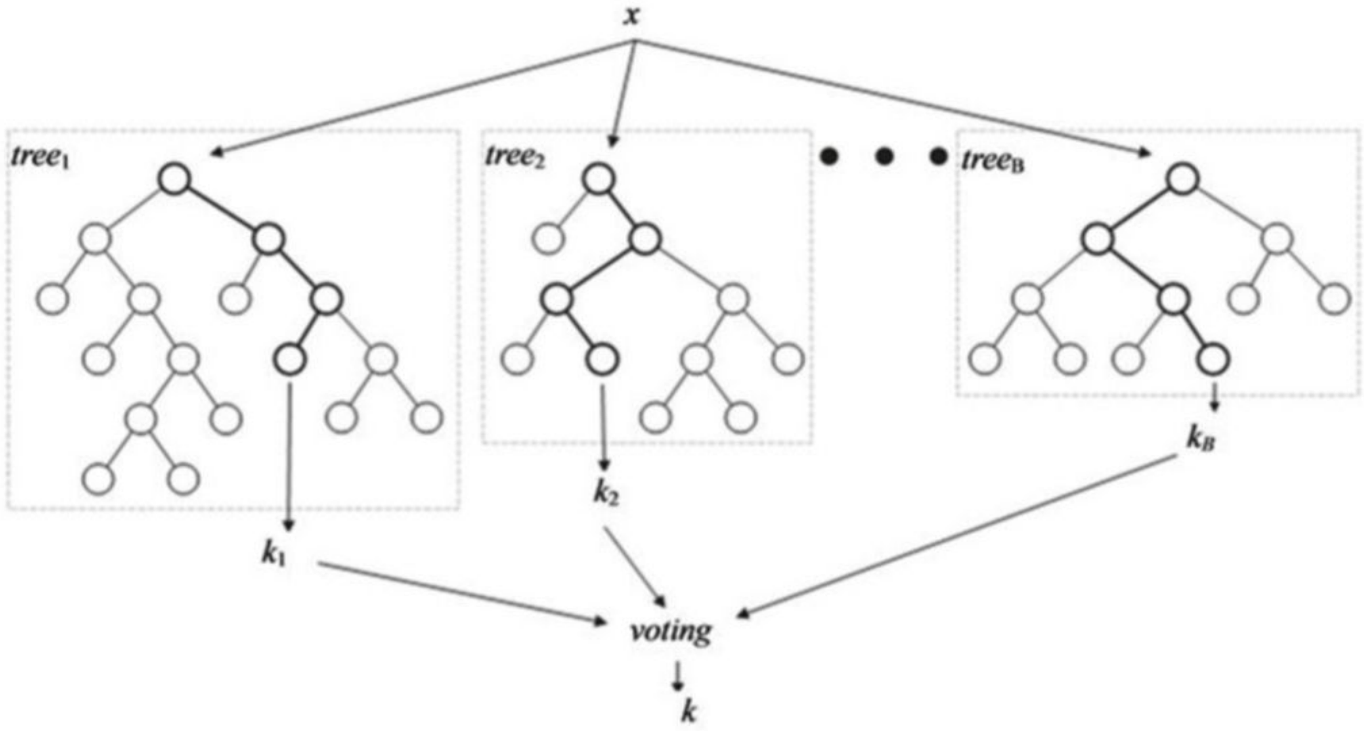
Higher entropy partitioning (large number of small partitions) is penalized!

TACC

Random Forest

- Ensemble of decision trees
- Each tree contains a random subset of **features**
- All trees are used to classify novel data (majority rule)

Random Forest



Return to your Jupyter Notebook

We'll continue with "SupervisedLearning.ipynb"

Break

we will reconvene at 2p CDT

Note: your VisPortal jobs will time out after <u>two hours</u> of runtime. Be sure to save any changes you want to keep. You can restart your session as needed.

Supervised Learning: Regression Methods

Regression

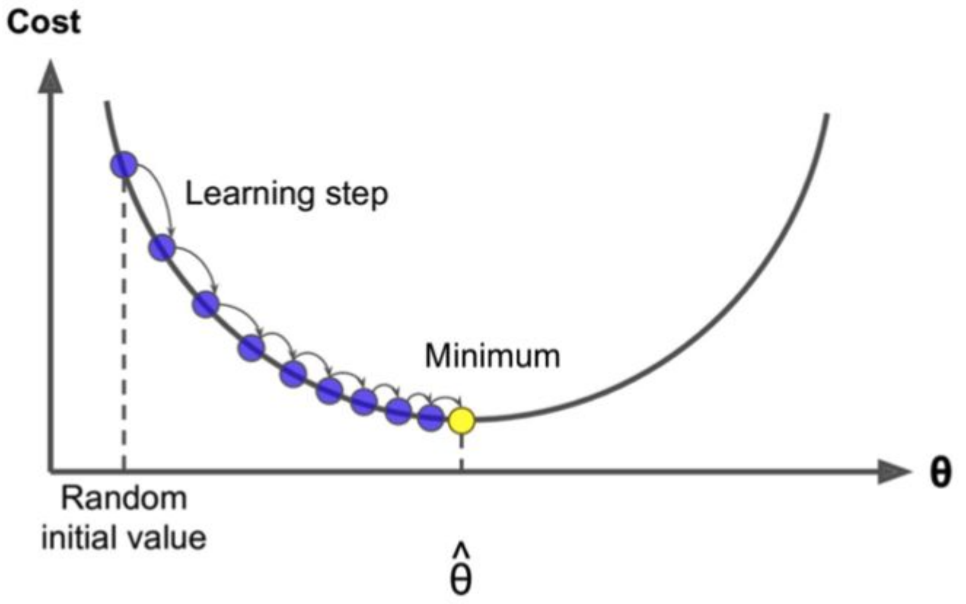
A statistical measure to determine the strength of the relationship between one **independent variable (x)** and one or more **dependent variables (y)** and **unknown parameters ($\boldsymbol{\theta}$)**.

$$y \sim F(x, \theta)$$

TACC

Gradient descent is used to fit parameters that minimize error

Given a cost function, select parameters $\boldsymbol{\theta}$ that minimize the cost



TACC

Commonly used cost functions

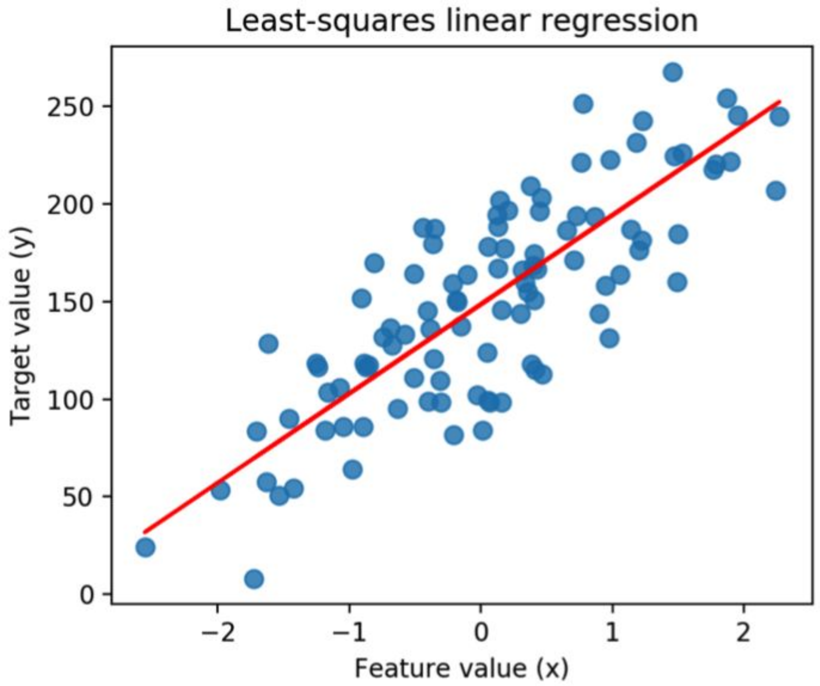
squared error: $\Sigma(y_{i,\ predicted} - y_{i,\ observed})^2$

least absolute deviation: $\Sigma|y_{i,\ predicted} - y_{i,\ observed}|$

Simple linear regression

One independent variable (y)
and one dependent variable (x)
connected through a function

$$y = ax + b$$



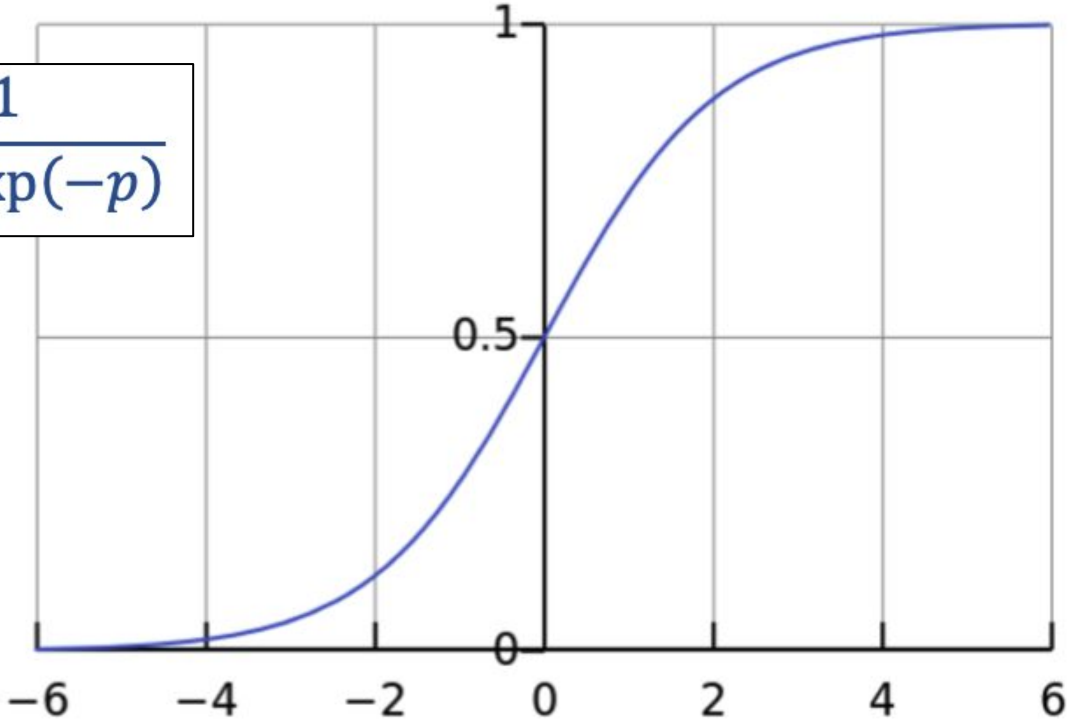
Binary logistic regression for classification

If **y** is a binary variable (positive/negative, pass/fail, healthy/sick), then **logistic regression** can predict the log(odds) of class outcome.

odds = Pr(outcome=A)/Pr(outcome=B)

Binary logistic regression for classification

$$logistic(p) = \frac{1}{1 + \exp(-p)}$$

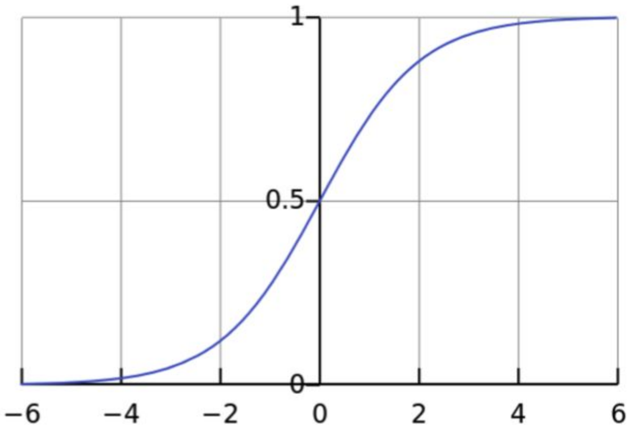


Binary logistic regression for classification

$$logistic(p) = \frac{1}{1 + \exp(-p)}$$

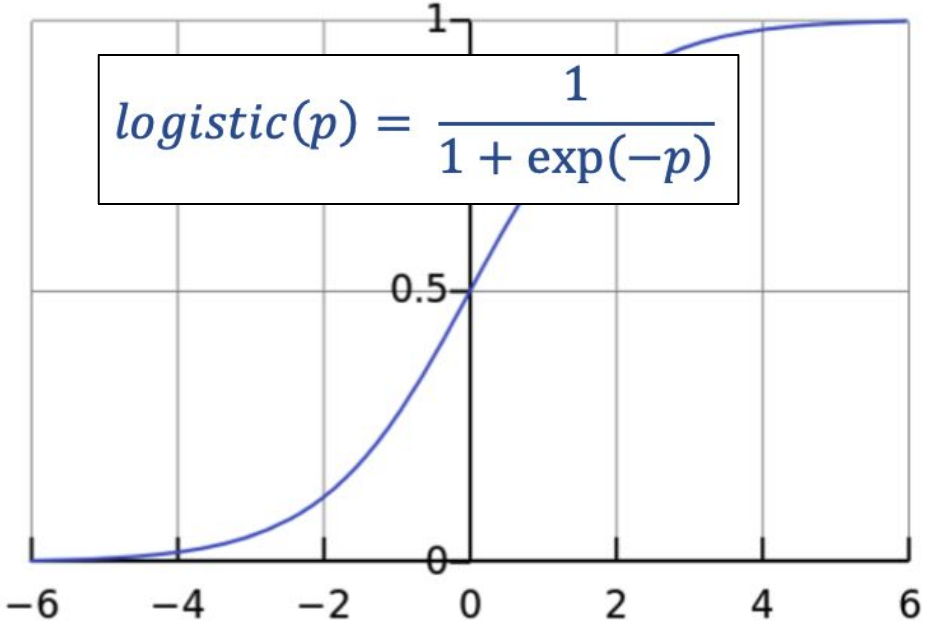
$$logit(p) = \ln\left(\frac{p}{1-p}\right) = ln\left(\frac{Pr(y=1)}{Pr(y=0)}\right)$$

$$ln\left(\frac{Pr(y=1)}{Pr(y=0)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$



Logistic (sigmoid) function is widely used

- logistic regression
- multinomial logistic regression
- activation function for neural networks



Back to regression

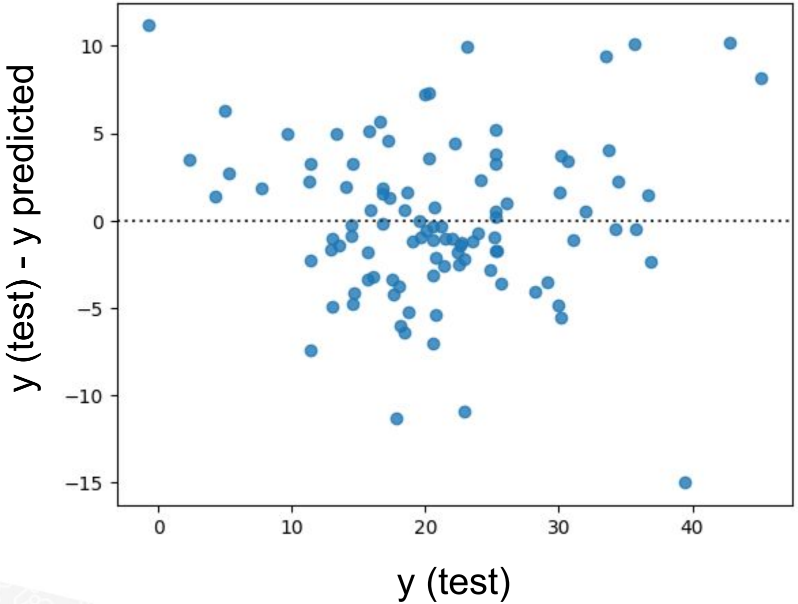
- Can regress on more than one variable

$$y = a_1x_1 + a_2x_2 + \ldots + a_nx_n + b$$

- If variables are correlated (e.g. temperature and humidity or weight and height), model fit may be suspect

Regression performance

Residual error should be normally distributed with mean zero



TACC

Return to your Jupyter Notebook

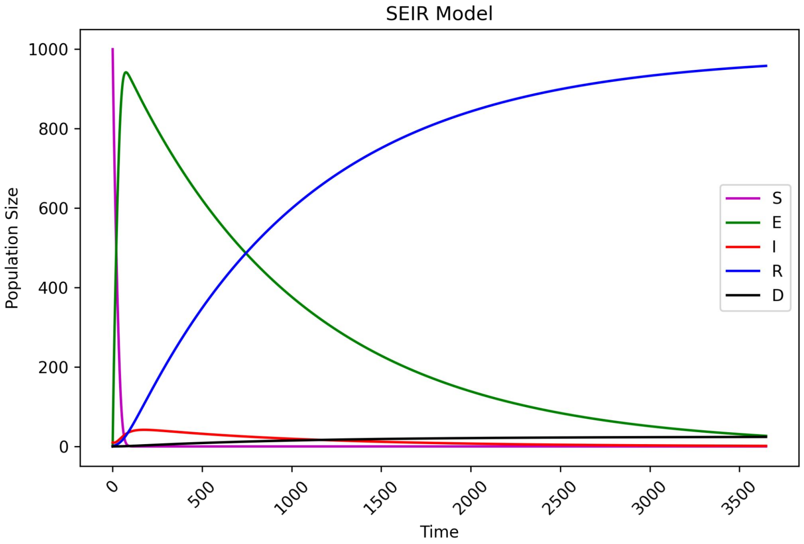
We'll continue with "SupervisedLearning.ipynb"

Non-linear regression for differential equation models

Population dynamics can be modeled with differential equations.

Disease models track populations of **susceptible** (S), **exposed** (E), **infectious** (I), **recovered** (R), and **deceased** (D) with systems of differential equations.

Regression can estimate rates in these model systems.



Return to your Jupyter Notebook

Open “ML_FitDiseaseModel.ipynb”