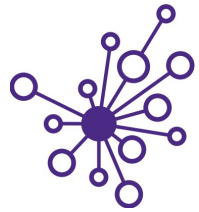
Knowledge and solutions for a changing world

Be boundless

Advancing data-intensive discovery in all fields



INTRODUCTION TO MACHINE LEARNING & KNN

UW DIRECT

(Data Intensive Research Enabling Cutting-edge Tech)

https://uwdirect.github.io

Stéphanie Valleau

Chemical Engineering

Outline

Introduction to Machine Learning (ML)

- Statistical learning – inference and prediction
- Parametric, non-parametric models
- Supervised, unsupervised models
- Regression, classification
- Error: Variance and Bias

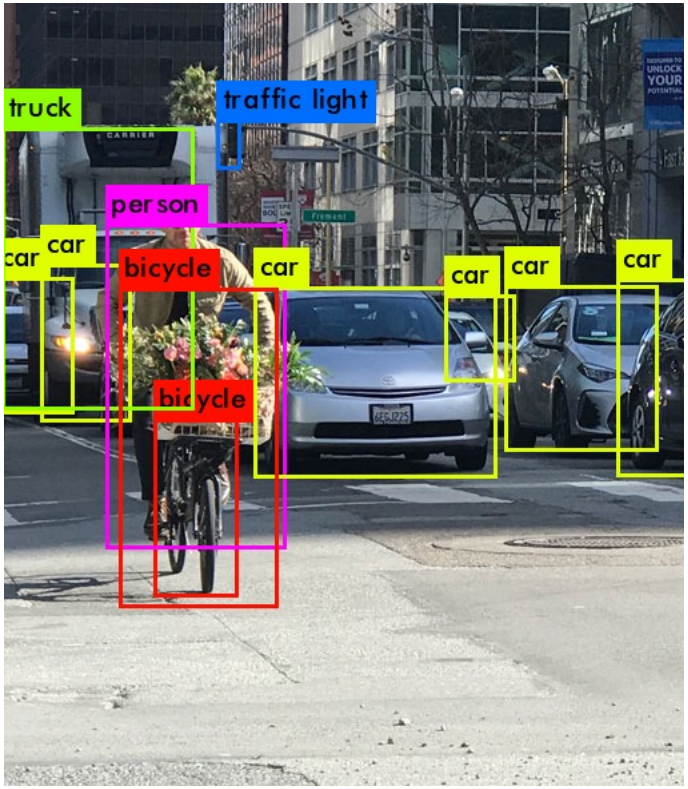
Classification methods

- K-nearest neighbors, KNN

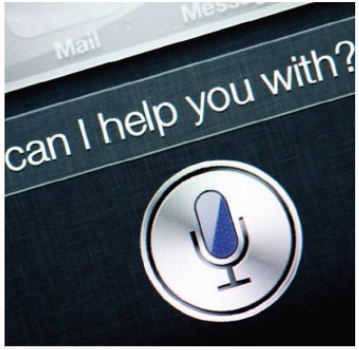
Many figures and notation will follow "*An Introduction to Statistical Learning: with applications in R*" unless otherwise noted; see the syllabus for a free PDF of the book.

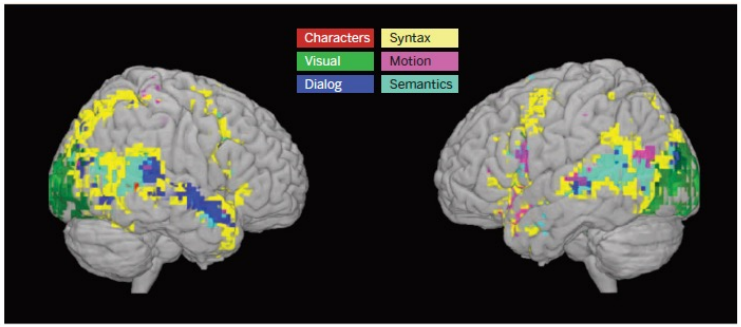
ML models can tackle many types of problems

Self-driving cars use trained ML algorithms to predict the objects that surround them



M. I. Jordan T. M. Mitchel Machine learning: Trends, perspectives, and prospects Science 255-260 (2015)





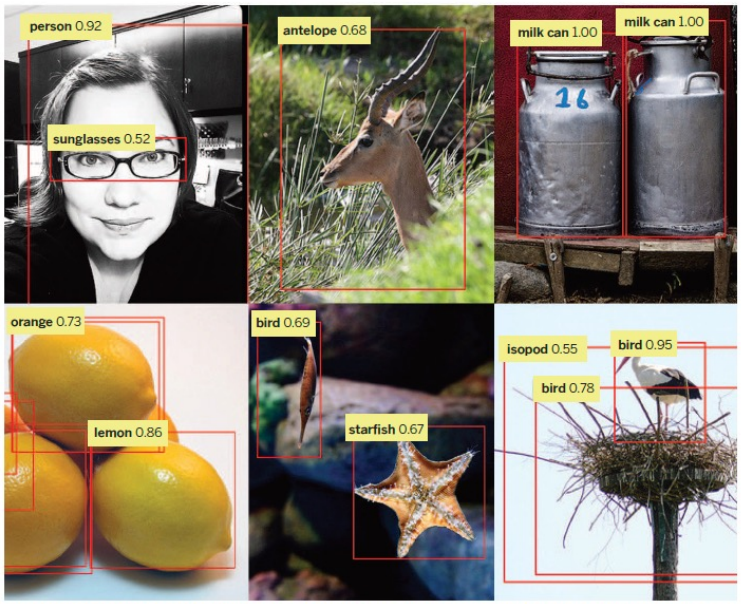


Fig. 1. Applications of machine learning. Machine learning is having a substantial effect on many areas of technology and science; examples of recent applied success stories include robotics and autonomous vehicle control (top left), speech processing and natural language processing (top right), neuroscience research (middle), and applications in computer vision (bottom). [The middle panel is adapted from (*29*). The images in the bottom panel are from the ImageNet database; object recognition annotation is by R. Girshick.]

Statistical learning

Ansatz

Given an input vector, X, and output vector, Y,
a function $f(X)$ exists such that

$$Y = f(X) + \epsilon$$

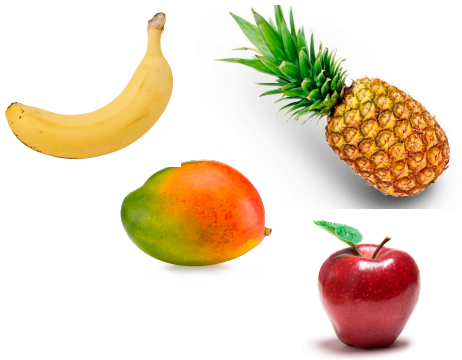
random error

unknown

$$X = \{X_1, X_2, \ldots, X_p\}$$

Why estimate *f*?

To predict the output *Y* (**prediction**)



e.g predict amount of fructose in an apple given its weight and type?

To connect X to Y (**inference**) – how do changes in X influence Y?

e.g. if the cost of tropical fruit increases will the amount of fruit sold go down?

Prediction

GOAL: estimate a function – an **estimator** - which gives the "best" prediction of *Y* given *X* in input

Hat symbol denotes estimator

$$\hat{Y} = \hat{f}(X)$$

How do we find it? How do we measure the **error?**

The **expected prediction error** of the estimator function is

$$\mathrm{E}\left(Y, \hat{f}(x)\right) = \mathrm{E}\left[\left(Y - \hat{f}(x)\right)^2\right] = \underbrace{\mathrm{E}\left[\left(f(x) - \hat{f}(x)\right)^2\right]}_{\text{reducible error}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{irreducible error}}$$

E – stands for the expectation value or mean squared error

How to estimate *f*?

Parametric vs non-parametric ML models

Parametric models

Model defined by functional form over the entire range of input features, X_i

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

Parameters:

$$\{\beta_0, \beta_1, \ldots, \beta_p\}$$

are found by minimizing the error between the model and exact values during training

Example: Linear regression

Non-parametric models

Do not assume a predefined functional form

There is still a nonparametric model fitting procedure

Require much larger training datasets

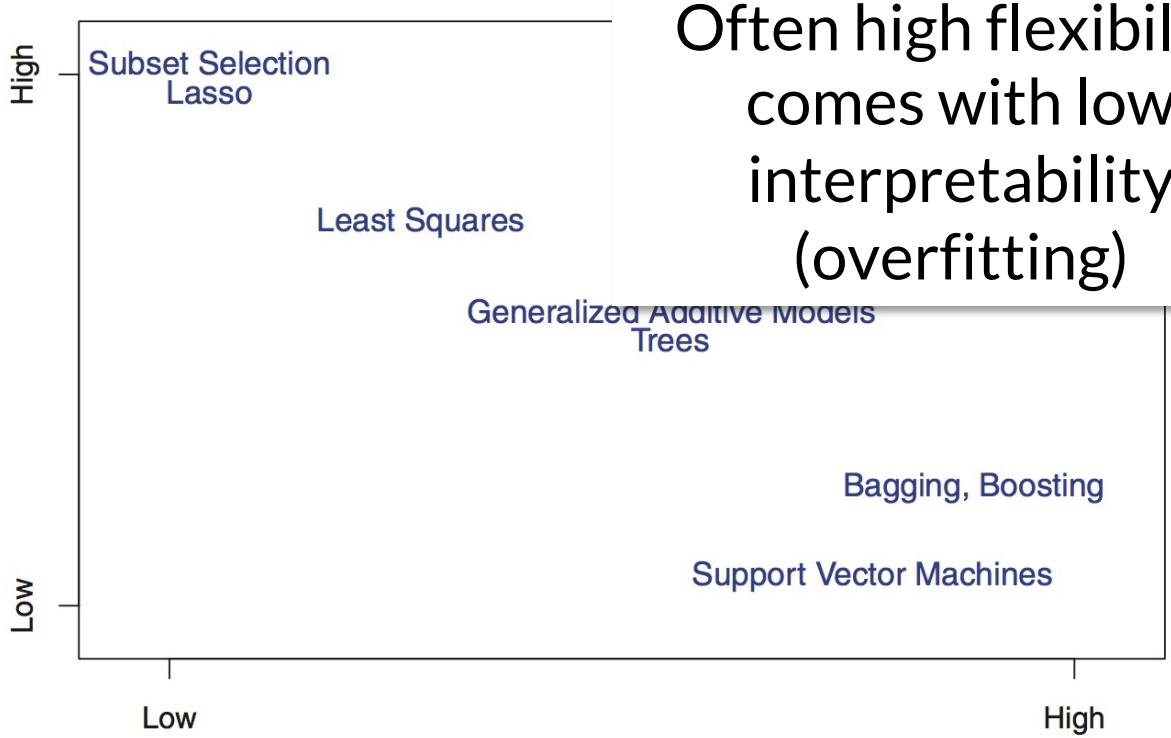
Example: KNN classification

Flexibility vs interpretability
The tradeoff

It all makes sense!

INTERPRETABILITY

Very confused



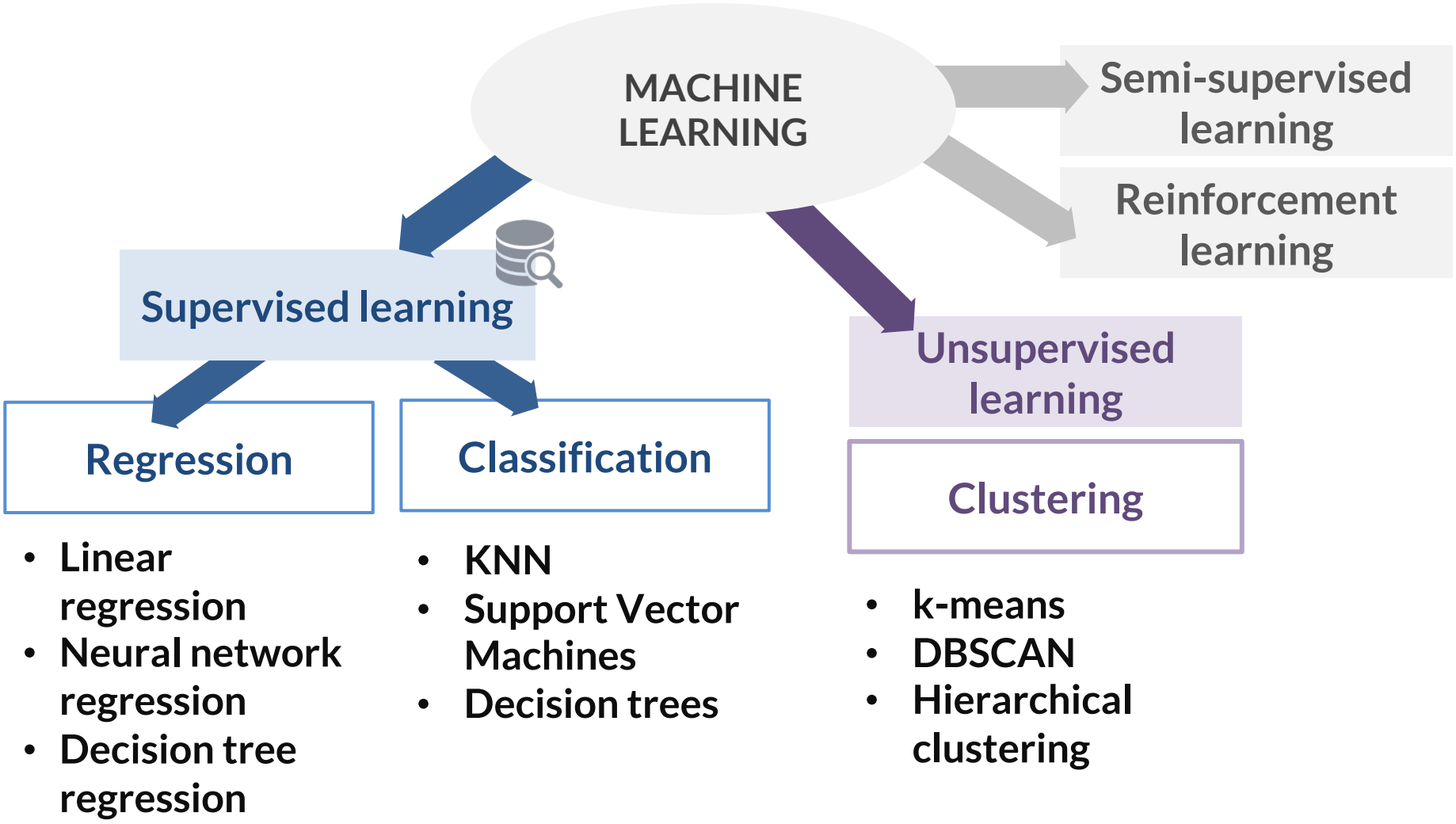
Often high flexibility comes with low interpretability (overfitting)

Eating apples must follow a linear trend during the day

FLEXIBILITY

With 10 million parameters I can predict all apple shapes in the world

Supervised vs Unsupervised



Supervised vs Unsupervised

***X* input feature/s – independent variables**
e.g. pictures, numbers
(weight / shape etc.)

SUPERVISED
ML models

PREDICT TARGET

target *Y* output
response / dependent variable
e.g. cost of fruit

***Unlabeled* *X* input feature/s**
e.g. pictures, numbers
(weight / shape etc.)

UNSUPERVISED
ML models

FIND PATTERNS

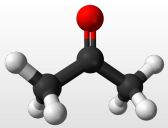
No predefined target Y !

e.g. find **clusters** in input data

Regression vs classification

Input: numeric or categorical

$$X = \{X_1, X_2, \ldots, X_N\}$$



Organic compound (categorical)
Atom positions (numeric)

ML MODEL

Output

$$\hat{Y} = \hat{f}(X)$$

quantitative
a.k.a. numerical

qualitative
a.k.a. categorical

Regression

$\hat{Y}$ numerical output:

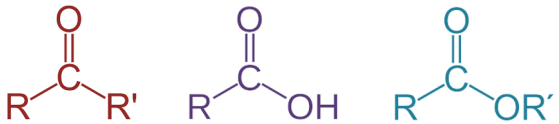
e.g. weight, energy (continuous)

m = 58.8 g/mol

Classification

$\hat{Y}$ categorical output:
category / class (discrete)

e.g. it is a ketone



Question

Think of an example of classification and / or regression & show it on your screen!

How accurate is my model?

Mean Squared Error (MSE) for regression

Training

Testing

$$X_{train} = \{x_1, x_2, x_3, \ldots, x_{N_{train}}\}$$

$$X_{test} = \{x_0, x_{N_{train}+1}, \ldots, x_{N_{test}}\}$$

Recall

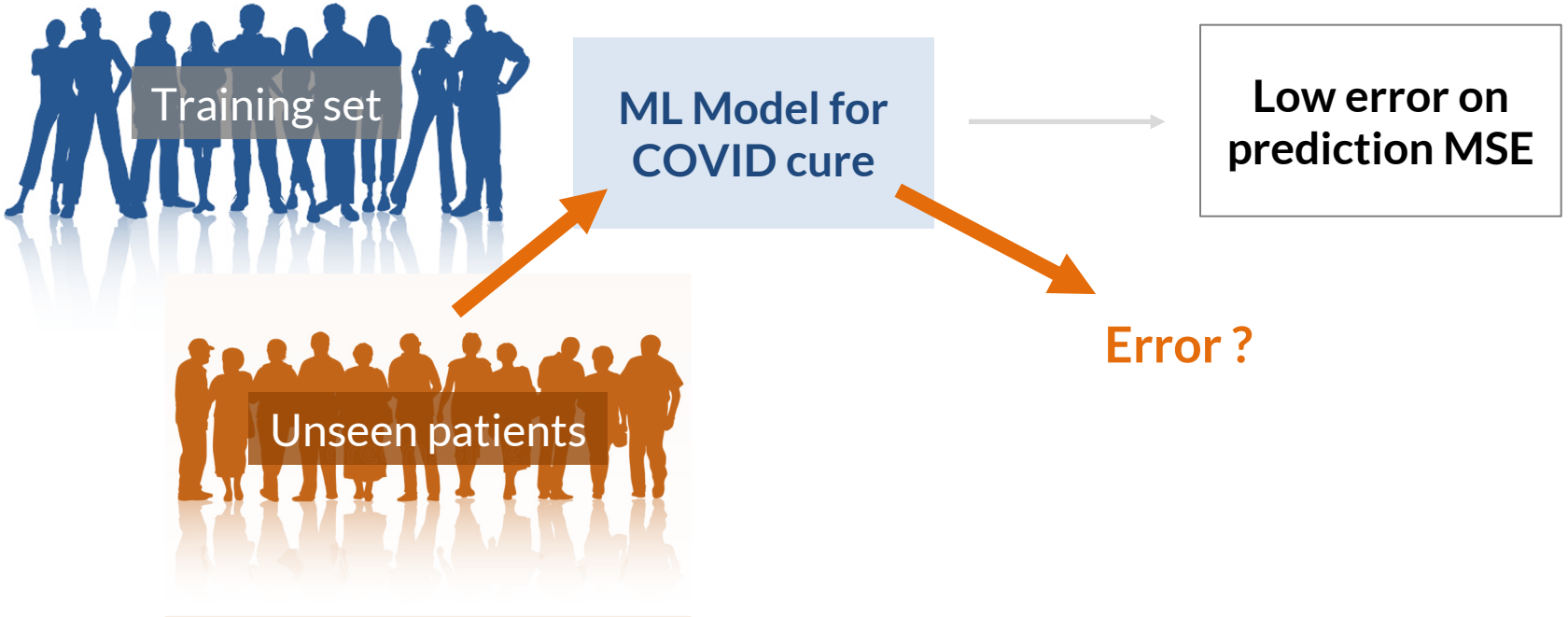
$$\mathrm{E}\left[\left(Y - \hat{f}(x)\right)^2\right] = \underbrace{\mathrm{E}\left[\left(f(x) - \hat{f}(x)\right)^2\right]}_{\text{reducible error}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{irreducible error}}$$

Why the test set?

$$\mathrm{MSE}_{test} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left(Y_i - \hat{f}(x_i)\right)^2 \equiv \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left(Y_i - \hat{Y}_i\right)^2$$

Assessing Model Accuracy

We always want to know how our model is doing "out in the real world" on some **test data** that the training process was blind to (validation) – We are not really interested in the training set.



Question

When would you trust the model if you were an unseen patient? Is a low error on the training set enough?

ML Model for COVID cure

Low error on prediction MSE

Error ?

Assessing Model Accuracy

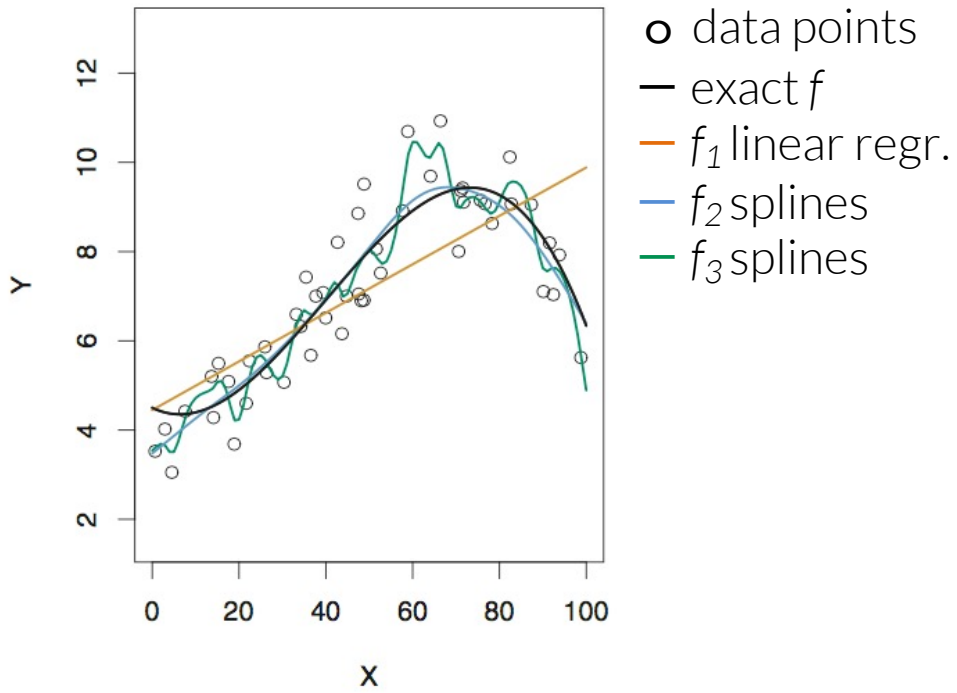
Trained ML Model for COVID cure

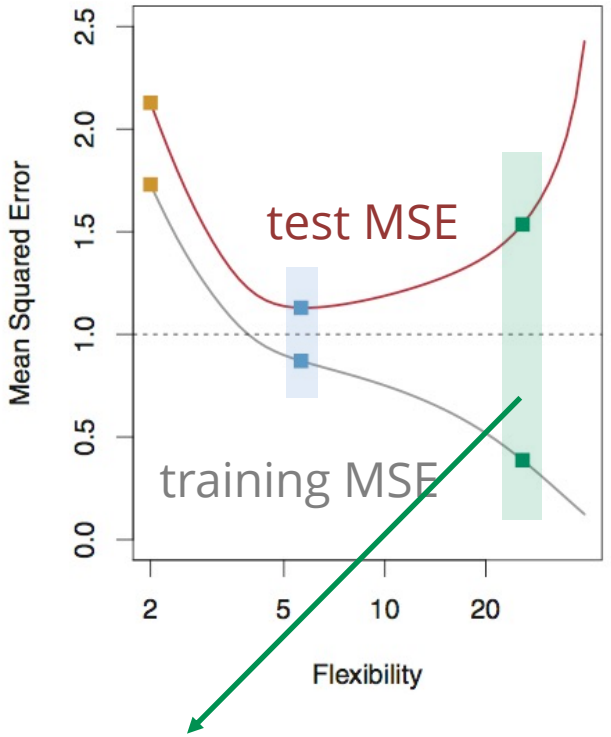
Low MSE on training set

Minimize error on the test set

We need both a low training set error **AND low test set error!!!**

Assessing Model Accuracy





Overfitting!

Hmm … why is the test curve U-shaped?

Fig 2.9, ISL

Potential pitfalls

Test set is **not representative** – with a different test set a much larger MSE!

Cross-validation will come to the rescue! (estimate test MSE using training set)

Variance and bias

How does the error change as I **change my test set**? How does it change with the type of model?

$$\mathrm{E}\left[(Y - \hat{f}(x))^2\right] = \underbrace{\mathrm{E}\left[(f(x) - \hat{f}(x))^2\right]}_{\text{reducible error}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{irreducible error}}$$

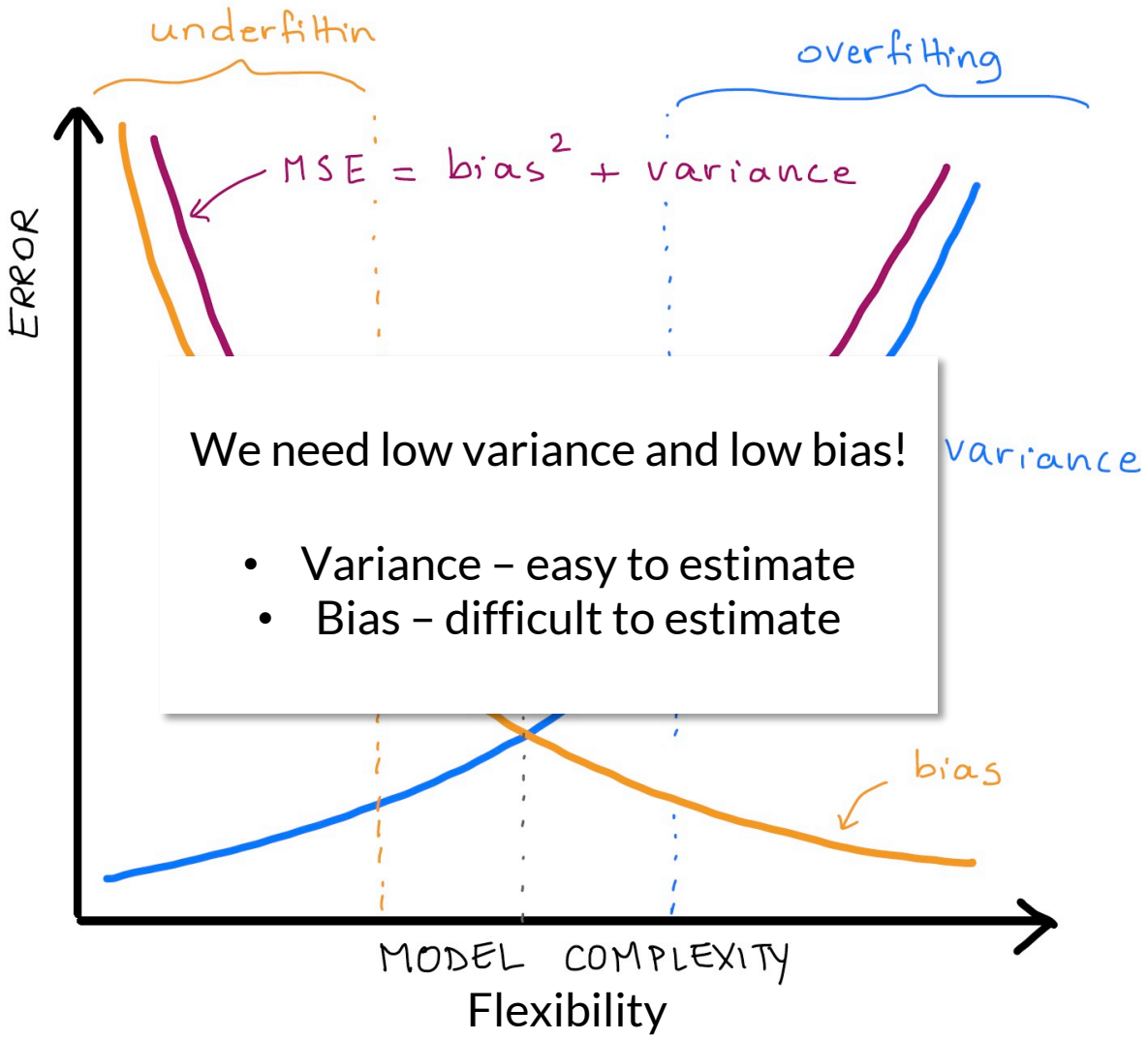
Average over training sets

$$E\left(y_{\text{test}} - \hat{f}(x_{test})\right)^2 = \mathrm{Var}\left(\hat{f}(x_{\text{test}})\right) + \left(\mathrm{Bias}(\hat{f}(x_{\text{test}}))\right)^2 + \mathrm{Var}(\epsilon)$$

Variance: quantifies how the choice of the **training set** influences $\hat{f}$

Bias: quantifies how the choice of the **model function** respect to the true f

Bias variance tradeoff



We need low variance and low bias!

- Variance – easy to estimate
- Bias – difficult to estimate

Question

Can the total error be zero?

Can I use MSE for classification?

Errors in classification problems

We are making qualitative predictions - how to assess the model accuracy?

We could look at the frequency of incorrect classification

$$\text{training error rate} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} I(y_i \neq \hat{y}_i)$$

0 if the two terms are equal;
1 otherwise

How can we use the test set? We want to minimize the error on the test set:

$$\mathrm{E}\left(I\left(y_{i,\text{test}} \neq \hat{y}_{i,\text{test}}\right)\right)$$

Estimating irreducible error in classification: Bayes classifier

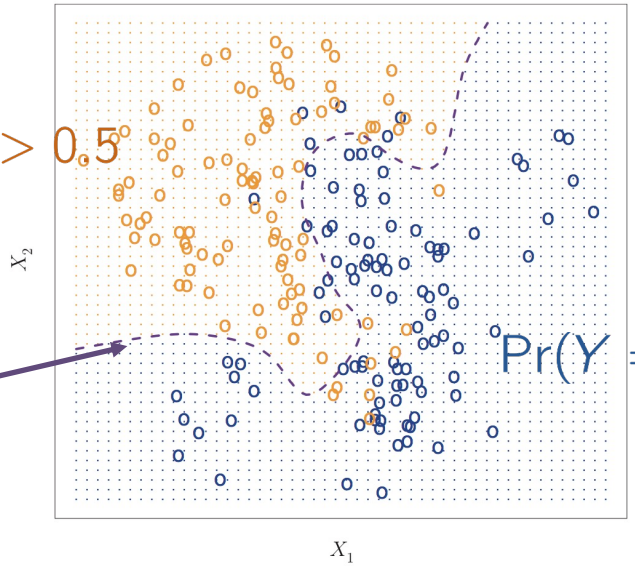
*Test error is minimized by the **Bayes classifier** that assigns each observation to most likely class given its predictor values*

Probability that Y belongs to class *j* given x_{test}

$$\Pr(Y = j \mid X = x_{test})$$

Observed predictor vector

$$\Pr(Y = \text{orange} \mid X = x_{test}) > 0.5$$



$$\Pr(Y = \text{orange} \mid X = x_{test}) < 0.5$$

Bayes decision boundary Pr = 0.5

$$\text{bayes error rate} = 1 - \mathrm{E}\left(\max_j \Pr(Y = j \mid X)\right)$$

One issue …

For real data we do not know the **conditional probability distribution** of *Y* given *X*!

Classification methods

Two types of classification methods

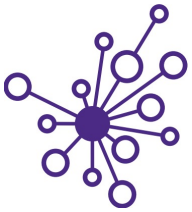
- ***k* Nearest Neighbors (often KNN)** – Making a map of parameter space and estimating the class membership based on *locality respect to other members of the class*
- **Logistic regression –** Making a mathematic guess about the *likelihood of class membership* using a simplified mathematical model (the logistic function)

$$\Pr(Y = j \mid X = x_{\text{test}})$$

KNN on Jupyter!

```
curl –O
https://raw.githubusercontent.com/UWDIRECT/UWDIRECT.github.io/master/Wi22_content/DSMCER/L6_Intro_ML_KNN.ipynb
```

More on KNN



After you have gone through the notebook **read the next few slides** – these provide information on the role of K

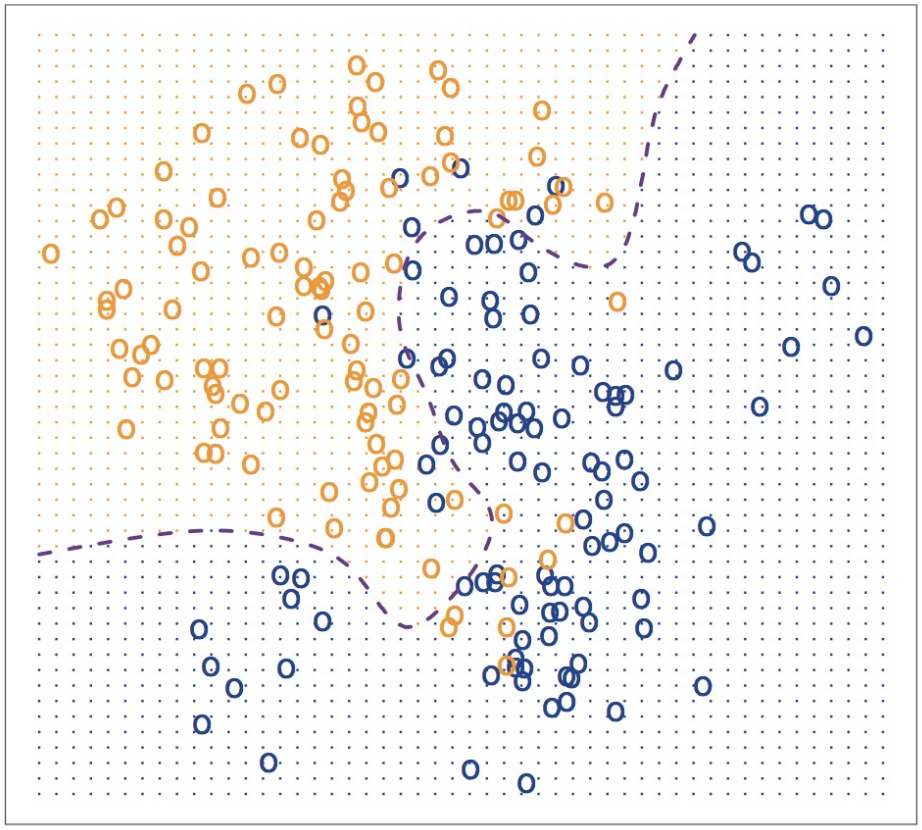
K-nearest neighbors (KNN)

The KNN classifier works on assigning the probability of membership in a class based on **distance** to other members in a class

- Our selection is based only on K – the number of neighbors we consider
- What are the possible ranges of K?
- Recall the KNN classifier

$$\Pr(Y = j \mid X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

A note about distances and identifying who is "nearest" ?



- The data in each dimension of X should be scaled similarly!
- We are concerned with relative distances in the parameter space, not absolute distances in the underlying units
- Otherwise, units, dimensionality, etc. can drastically favor closeness in arbitrary dimensions of X

Illustration of KNN

$$\Pr(Y = j \mid X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

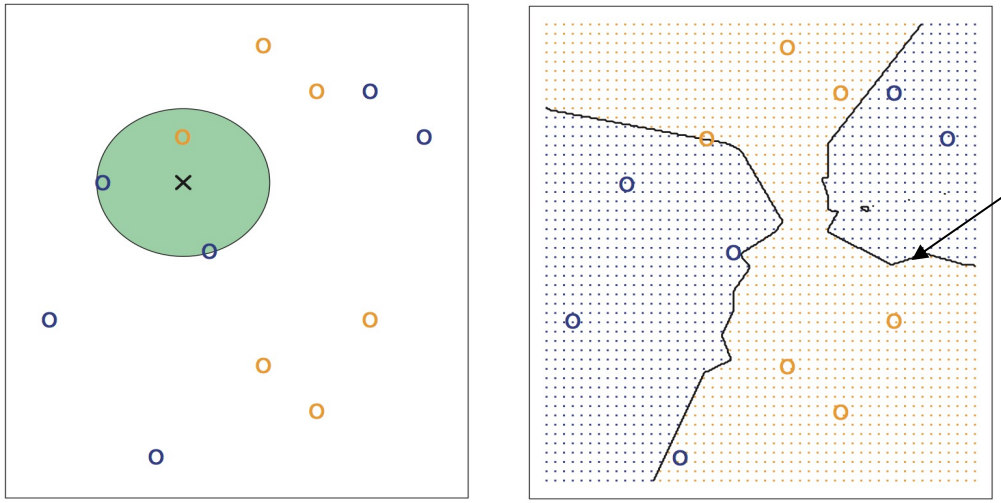
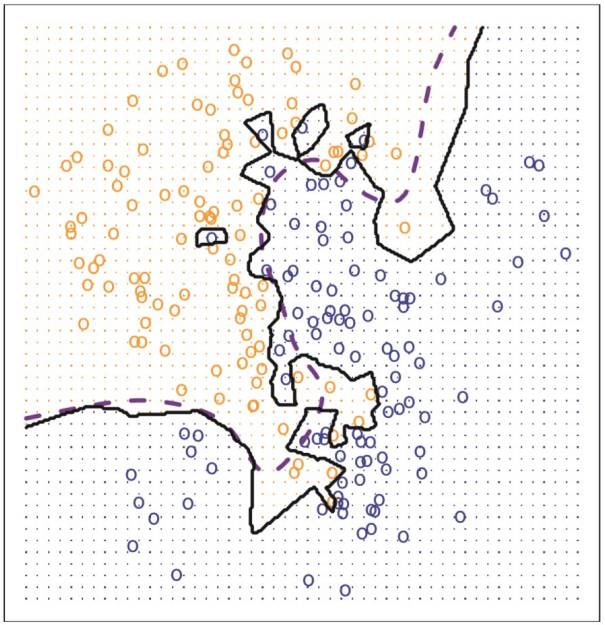


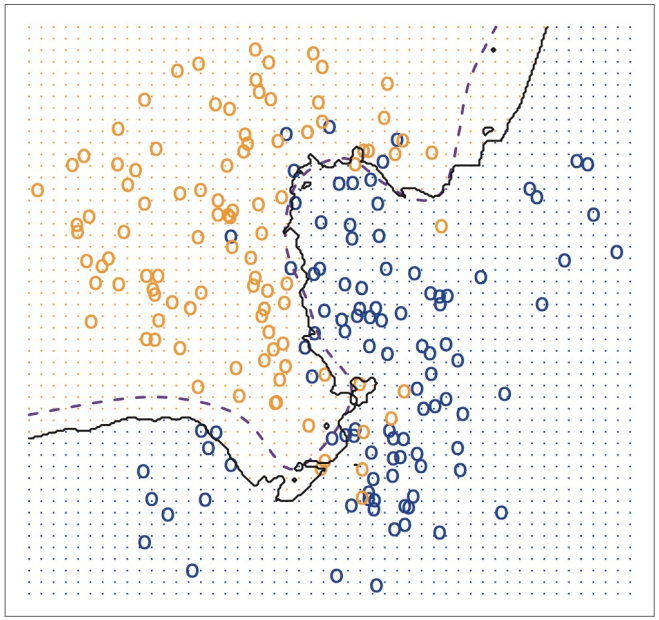
FIGURE 2.14. *The KNN approach, using $K = 3$, is illustrated in a simple situation with six blue observations and six orange observations.* Left: *a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue.* Right: *The KNN decision boundary for this example is shown in black. The blue grid indicates the region in which a test observation will be assigned to the blue class, and the orange grid indicates the region in which it will be assigned to the orange class.*

Effect of K on boundary

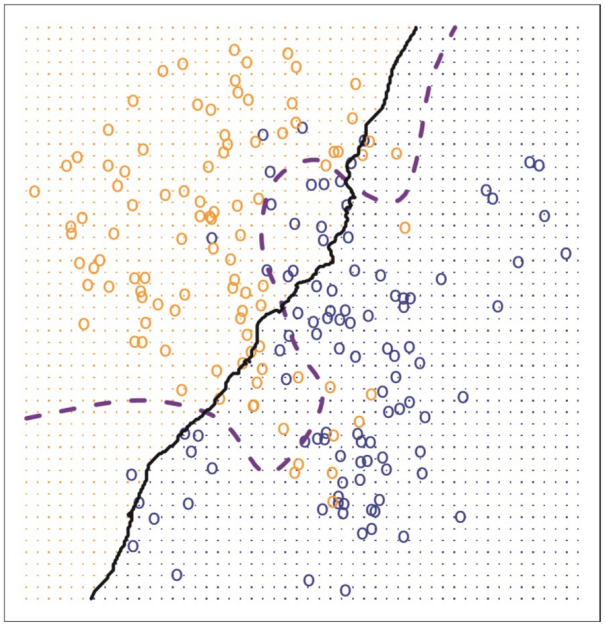
K=1



K=10



K=100



High flexibility

Low flexibility

Low bias

Low variance

Effect of K on accuracy

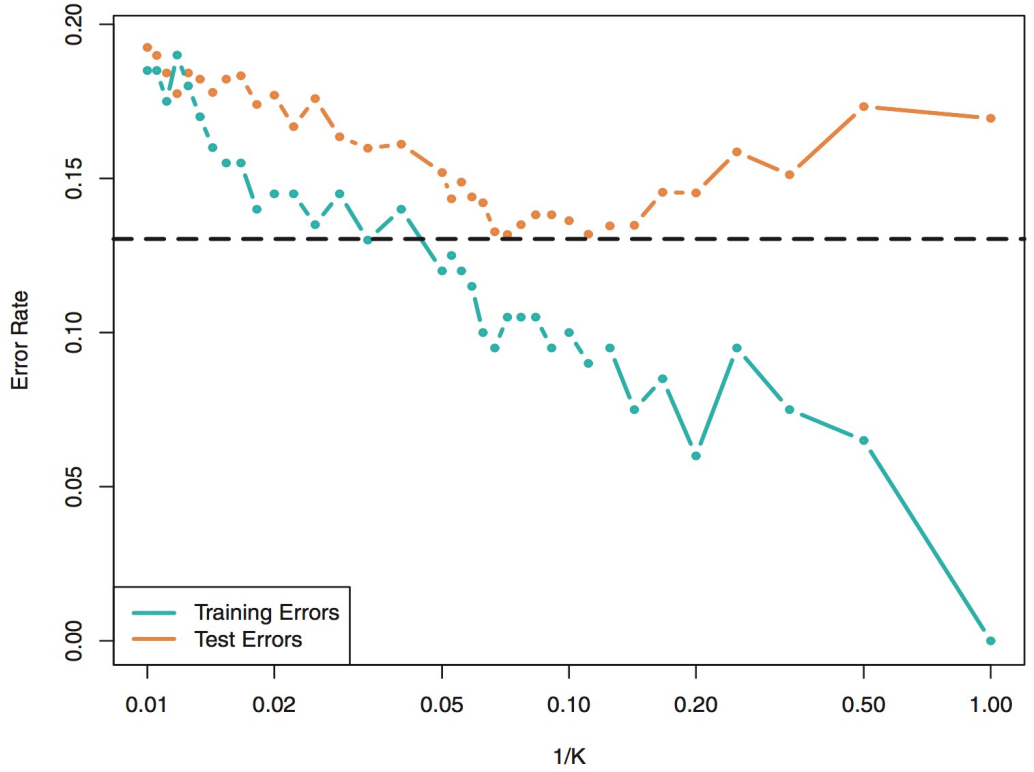


FIGURE 2.17. *The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the data from Figure 2.13, as the level of flexibility (assessed using $1/K$) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training data set.*