Imperial College
Business School

Imperial means
Intelligent Business

Big Data in Finance

Introduction to Machine Learning

Tarun Ramadorai

Administrative Notes

- Who am I? My Office Hours: Wednesdays 5 to 6 pm.
- TAs: Qing Yao and Chao Zhang: q.yao15@imperial.ac.uk; chao.zhang113@imperial.ac.uk.
- All materials will be on the hub at least one week prior to the next lecture.
- Please look at this at least a few days prior to every lecture, and read the appropriate starred (*) papers/chapters.

Assessment

- *Group Assignment 1*. 25% of overall course grade.
  - Handed Out: Friday, 20th January.
  - Due: Friday 10th February.

- *Group Assignment 2*. 15% of overall course grade.
  - Handed Out: Friday, 10th February.
  - Due: Friday 3rd March, In-class presentation.

- *Final examination*. 60% of overall course grade.
  - Between 13-24 March.

Introduction

- "There was five exabytes of information created between the dawn of civilization through 2003, but that much information is now created every two days, and the pace is increasing." – Eric Schmidt, former CEO of Google, 2010.

Introduction

- "There was five exabytes of information created between the dawn of civilization through 2003, but that much information is now created every two days, and the pace is increasing." – Eric Schmidt, former CEO of Google, 2010.

- "Big Data is Not About the Data!" – Gary King, Institute for Quantitative Social Science, Harvard.

Introduction

- "There was five exabytes of information created between the dawn of civilization through 2003, but that much information is now created every two days, and the pace is increasing." – Eric Schmidt, former CEO of Google, 2010.

- "Big Data is Not About the Data!" – Gary King, Institute for Quantitative Social Science, Harvard.

- "I keep saying that the sexy job in the next 10 years will be statisticians, and I'm not kidding." – Hal Varian, Chief Economist at Google.

Introduction

- "There was five exabytes of information created between the dawn of civilization through 2003, but that much information is now created every two days, and the pace is increasing." – Eric Schmidt, former CEO of Google, 2010.

- "Big Data is Not About the Data!" – Gary King, Institute for Quantitative Social Science, Harvard.

- "I keep saying that the sexy job in the next 10 years will be statisticians, and I'm not kidding." – Hal Varian, Chief Economist at Google.

- "If we have data, let's look at data. If all we have are opinions, let's go with mine." – Jim Barksdale, former Netscape CEO.

Introduction

- "There was five exabytes of information created between the dawn of civilization through 2003, but that much information is now created every two days, and the pace is increasing." – Eric Schmidt, former CEO of Google, 2010.

- "Big Data is Not About the Data!" – Gary King, Institute for Quantitative Social Science, Harvard.

- "I keep saying that the sexy job in the next 10 years will be statisticians, and I'm not kidding." – Hal Varian, Chief Economist at Google.

- "If we have data, let's look at data. If all we have are opinions, let's go with mine." – Jim Barksdale, former Netscape CEO.

- "Torture the data, and it will confess to anything." – Ronald Coase, Economics Nobel Prize Laureate.

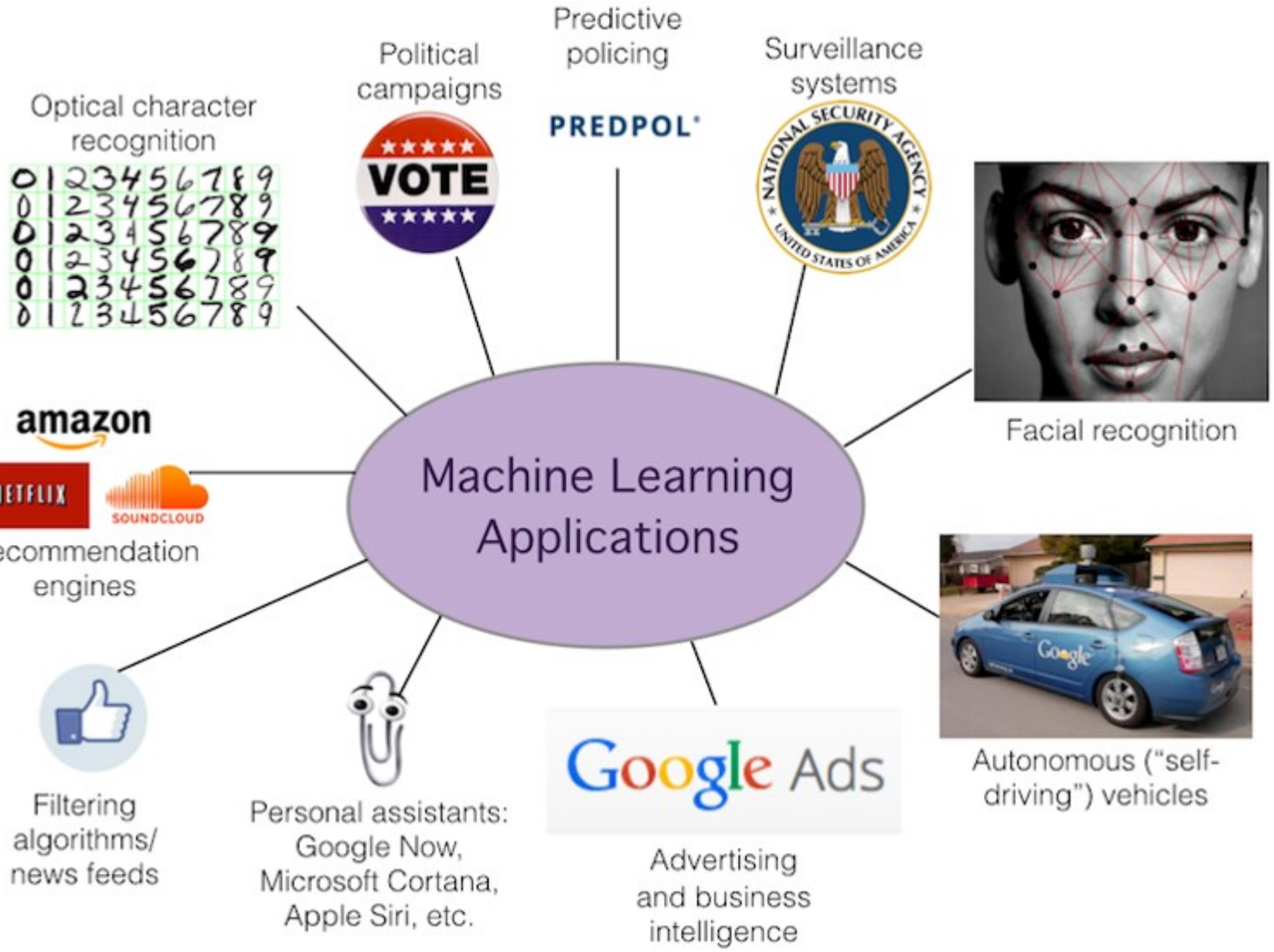
New Datasets in Finance

- Unstructured text: social media, web pages, news, court judgements, company annual reports.
- Financial : company balance sheets, house prices, stock prices, forex rates.
- Geographic: postcodes/zipcodes and all associated information, as well as satellite imagery.
- Electoral activity: electoral rolls, voting records.
- Web: clicks, searches, advertising clickthroughs, supermarket prices.
- And many more…

Machine Learning

- The data is ubiquitous, growing rapidly, and increasingly, publicly available.
- Scarce resource is understanding and skillful utilization of appropriate techniques for analysis.
- In this course, we will spend time developing an understanding of a variety of techniques for both prediction and inference in large datasets. Note: These will mainly be (but are not limited to) machine learning techniques.
- Focus is primarily outcome oriented/applied rather than theoretical, but we will consider simple theoretical foundations where necessary (mainly today).
- We begin today with an introduction to some machine learning tools that we will be using in the course, alongside some examples. Note: This is probably the most theoretical lecture we will have!

Some non-Finance Applications of Machine Learning



A Few Basic Machine Learning Insights

- ML has a reasonably long history, and one aspect of this history is development of techniques in response to specific problems. This primarily applied focus means:

- Less reliance on formal statistical (asymptotic) theory.

A Few Basic Machine Learning Insights

- ML has a reasonably long history, and one aspect of this history is development of techniques in response to specific problems. This primarily applied focus means:

- Less reliance on formal statistical (asymptotic) theory.

- Focus on “does it work?” rather than “why does it work?”.

A Few Basic Machine Learning Insights

- ML has a reasonably long history, and one aspect of this history is development of techniques in response to specific problems. This primarily applied focus means:

- Less reliance on formal statistical (asymptotic) theory.

- Focus on "does it work?" rather than "why does it work?".

- Methods for evaluation of "does it work" quite different from (perhaps) more familiar statistical approaches (e.g., cross-validation).

A Few Basic Machine Learning Insights

- ML has a reasonably long history, and one aspect of this history is development of techniques in response to specific problems. This primarily applied focus means:

- Less reliance on formal statistical (asymptotic) theory.

- Focus on "does it work?" rather than "why does it work?".

- Methods for evaluation of "does it work" quite different from (perhaps) more familiar statistical approaches (e.g., cross-validation).

- Great at prediction, but only now moving into inference and causality.

A Few Basic Machine Learning Insights

- ML has a reasonably long history, and one aspect of this history is development of techniques in response to specific problems. This primarily applied focus means:

- Less reliance on formal statistical (asymptotic) theory.

- Focus on "does it work?" rather than "why does it work?".

- Methods for evaluation of "does it work" quite different from (perhaps) more familiar statistical approaches (e.g., cross-validation).

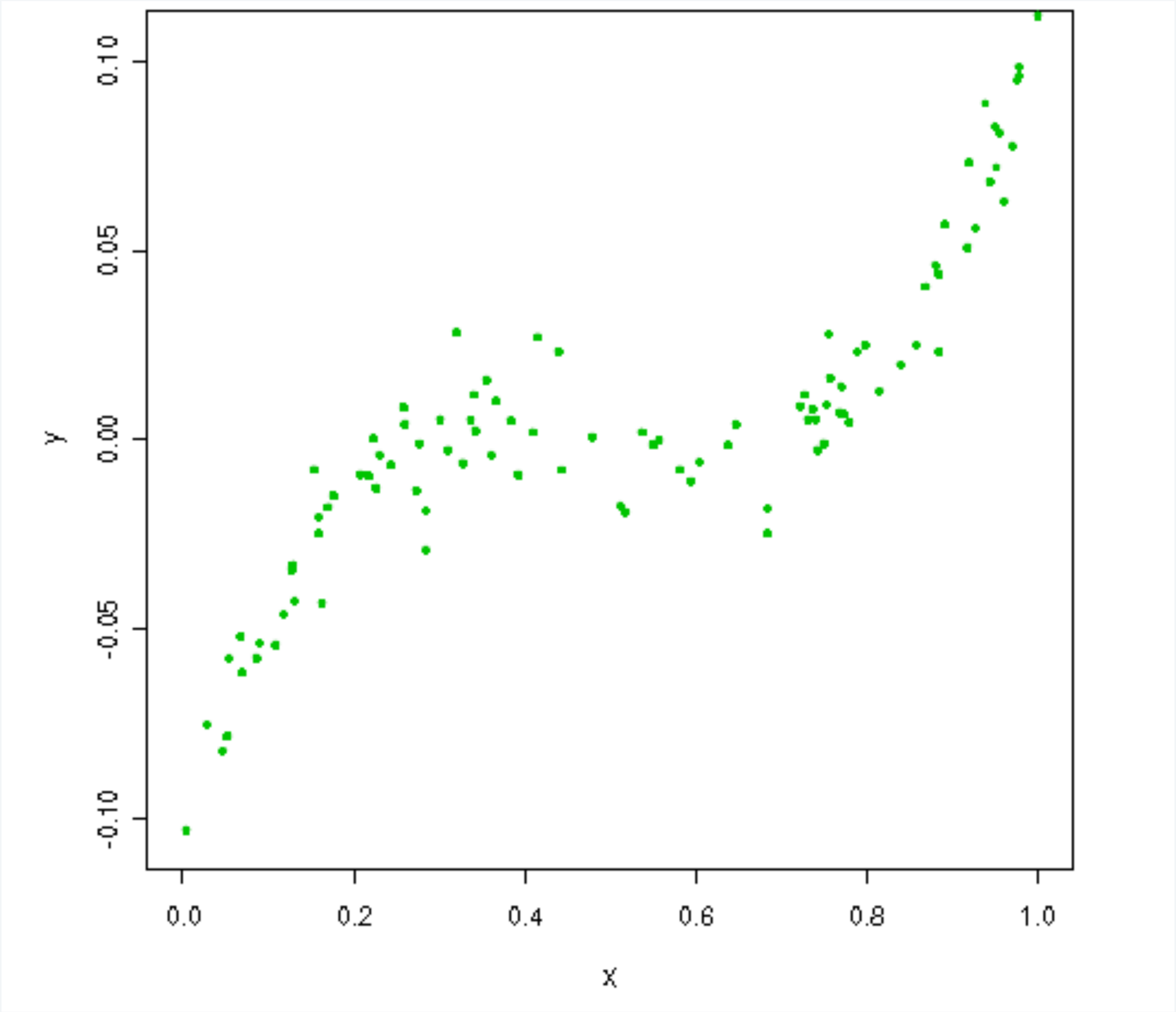
- Great at prediction, but only now moving into inference and causality.

- Note: Jargon is also different (AUC, ROC, Training and Test, Hold-Out, K-Fold, (Un)supervised, Tuning, Bagging, Boosting, Regularization…).

A Simple Introduction to Statistical Learning

- Suppose we observe Y_i and $X_i = (X_{i1}, ..., X_{ip})$ for $i = 1, ..., n$.
- And suppose that we believe that there is a relationship between Y and at least one of the X's.
- Then we can model the relationship as $Y_i = f(X_i) + \varepsilon_i$, where f is an unknown function and ε is a random error term with mean zero.
- Statistical learning is the science (and art) of discovering $f(.)$
  - Called learning, because we are using the data to "learn" f.
- Learning f will be useful for two main purposes:
  - Prediction: What's going to happen (given new X)?
  - Inference: Which predictors are critical and why?

The Sort of Thing we are Trying to Learn



Classification and Regression

- One example of supervised learning (note: what is unsupervised learning?) is classification:
- Suppose you have N observations on pairs (Y_i, X_i), where Y_i is an element of a (not necessarily ordered) discrete-valued set $\{0, 1, ..., J\}$.
- Goal is to find a function $f(x; X, Y)$ that assigns a new observation with $X = x$ to one of the categories (less interest in probabilities, more in actual assignment).
- Assume x is a draw from the same distribution as $X_i, i = 1, ..., N$.
- Big success: automatic reading of zipcodes; classify each handwritten digit into one of ten categories; face recognition in pictures; spam filtering.
- Less analysis of causality, mostly pure prediction. One finance application: Credit default forecasting..

Machine Learning vs Traditional Credit Scoring

Khandani, Kim, Lo (2010)

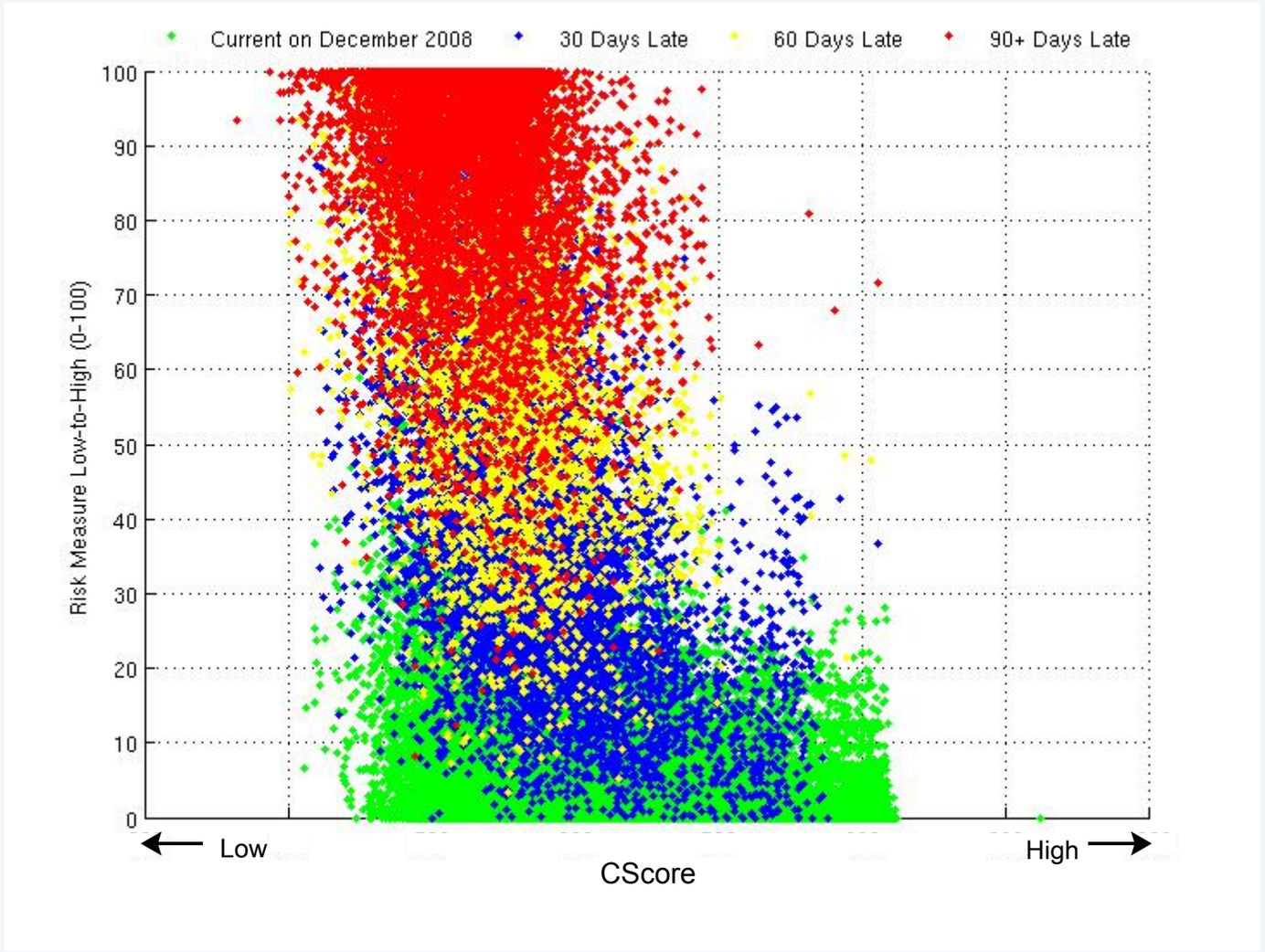
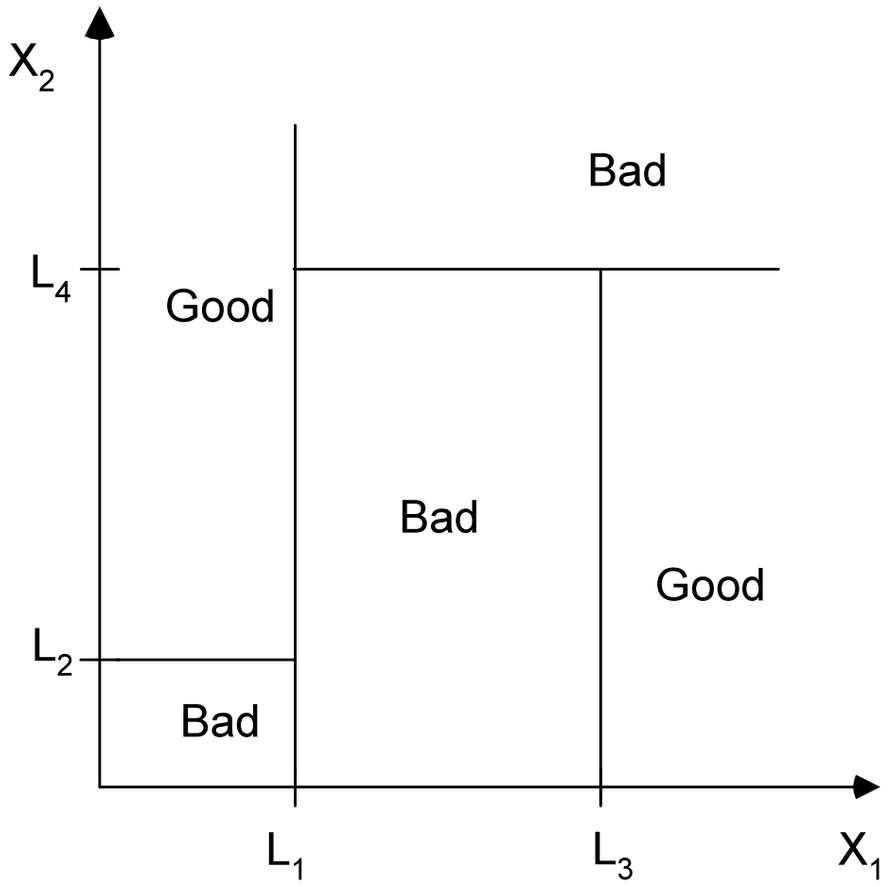
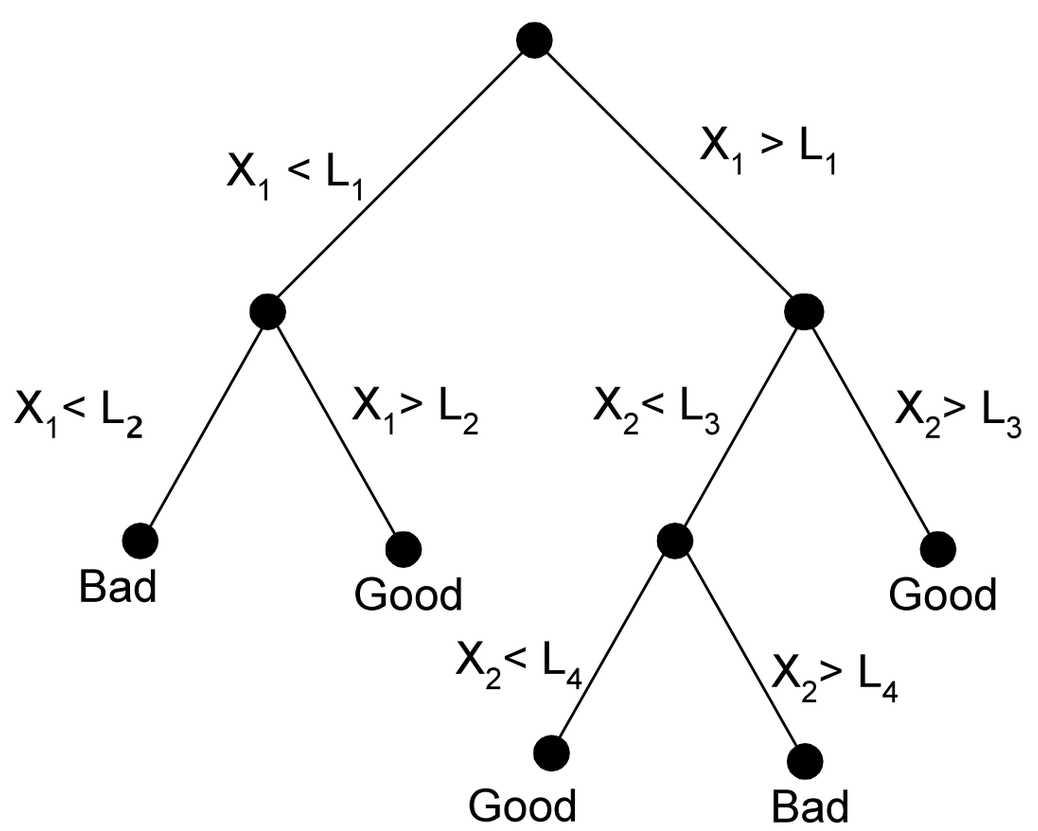


Figure 13: Color-coded comparison of December 2008 CScore (x-axis, and where higher values are associated with lower credit risk) and machine-learning forecasts of 90-days-or-more delinquency rates over subsequent 3-month windows using realized delinquency events from October to December 2008 and September 2008 feature vectors. Once calibrated, the machine-learning model is applied to the December 2008 feature vector, and the resulting "fitted values" are plotted (y-axis) against the December 2008 CScores. The color coding indicates whether an account is current (green), 30-days delinquent (blue), 60-days delinquent (yellow), or 90-days-or-more delinquent as of December 2008.

Tree Model Schematic
Khandani, Kim, Lo (2010)





Classification and Regression

- One (possibly familiar) example from econometrics is nonparametric regression (think sorting). There are many cases where we simply need a good fit for the conditional expectation.

- N observations on pairs (Y_i, X_i). Y_i continuous rather than discrete. Goal is to find a function $f(x; X, Y)$ that is a good predictor for Y for a new observation $X = x$.
  - For example, kernel regression is widely used in econometrics, but less useful when x is high-dimensional:

$$f(x|X,Y) = \frac{\sum_{i=1}^{N} Y_i K\left(\frac{x-X_i}{h}\right)}{\sum_{i=1}^{N} K\left(\frac{x-X_i}{h}\right)}$$

- One finance application of regression: Equity premium prediction..

Classification and Regression

- One (possibly familiar) example from econometrics is nonparametric regression (think sorting). There are many cases where we simply need a good fit for the conditional expectation.

- N observations on pairs (Y_i, X_i). Y_i continuous rather than discrete. Goal is to find a function $f(x; X, Y)$ that is a good predictor for Y for a new observation $X = x$.
  - For example, kernel regression is widely used in econometrics, but less useful when x is high-dimensional:

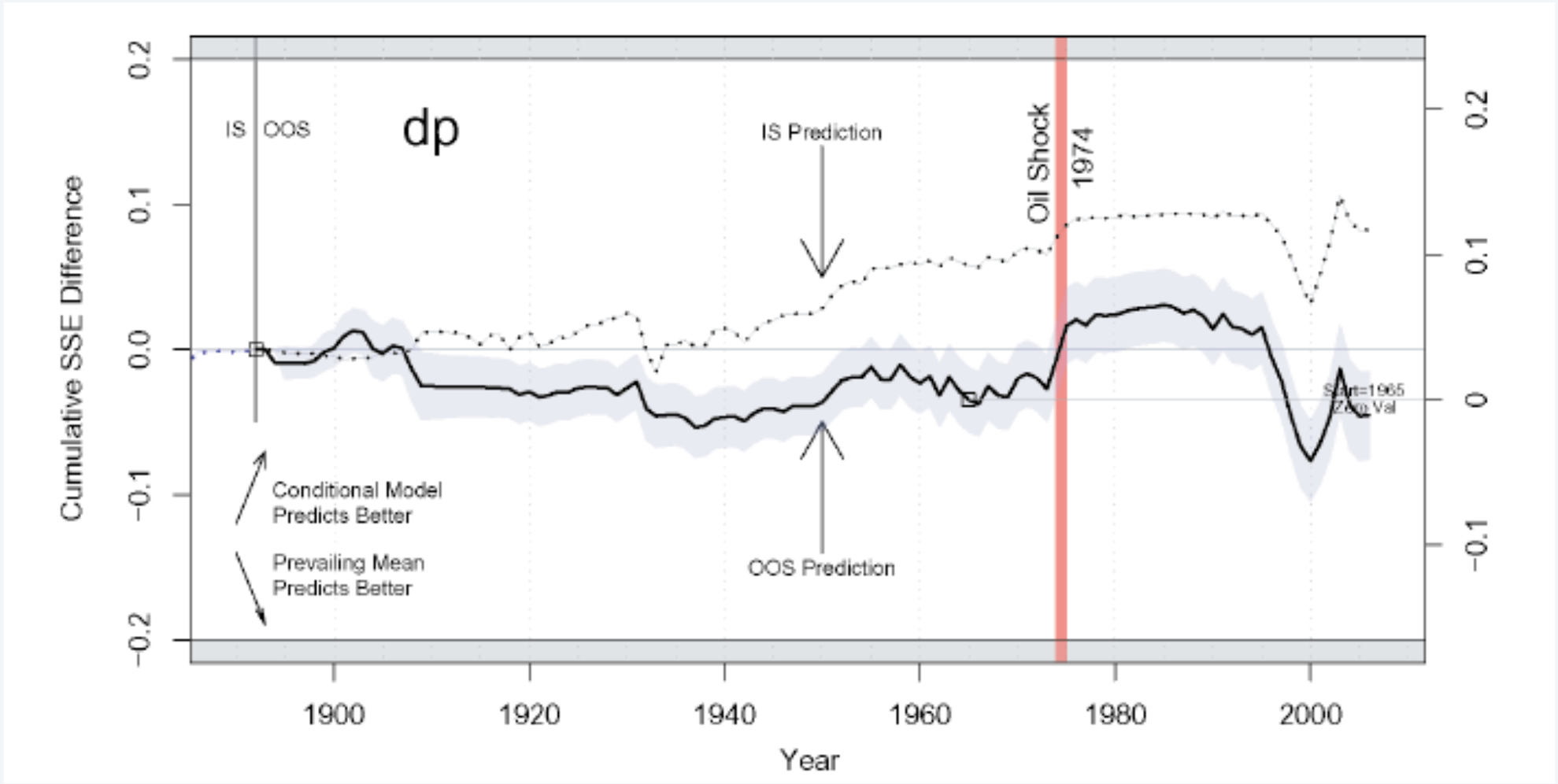
$$f(x|X,Y) = \frac{\sum_{i=1}^{N} Y_i K\left(\frac{x-X_i}{h}\right)}{\sum_{i=1}^{N} K\left(\frac{x-X_i}{h}\right)}$$

- One finance application of regression: Equity premium prediction..

- We will also discuss how machine learning can help with (especially causal) inference later in the course. Currently a hugely active area of research, and a very exciting direction.

Sneak Peek: Equity Premium

Goyal and Welch (2008)



Validation

- Different methods are generally validated by assessing their properties out of sample.

- This is much easier for prediction problems than for causal problems.
  - For prediction problems we see realizations, so that a single observation can be used to estimate the quality of the prediction.
  - A single realization of (Y_i, X_i) gives us an unbiased estimate of $\mu(x) = E[Y_i|X_i = x]$, namely Y_i.

- For causal problems we do not generally have unbiased estimates of the true causal effects (think of omitted variable issues, for example).

- Usual approach is to use a "training sample" to "train" (estimate) model, and a "test sample" to compare and evaluate algorithms.

Validation

- How might we "train" any new model on a training sample?
- General approach is to tune the model (or pick amongst models) to minimize the residual sum of squares on the training sample:

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

- For a classification problem, we can use the error rate:

$$ErrorRate = \frac{1}{N} \sum_{i=1}^{N} I(y_i \neq \hat{y}_i)$$

Validation

- However, you might be starting to worry that optimization on the training sample is somewhat misleading.

- What we really care about is how well any method will work on data we haven't seen, i.e., on a "test sample."

- No guarantee that low training sample error rates lead to low test sample error rates...
  - Indeed, generally negatively correlated!

The Bias-Variance Tradeoff

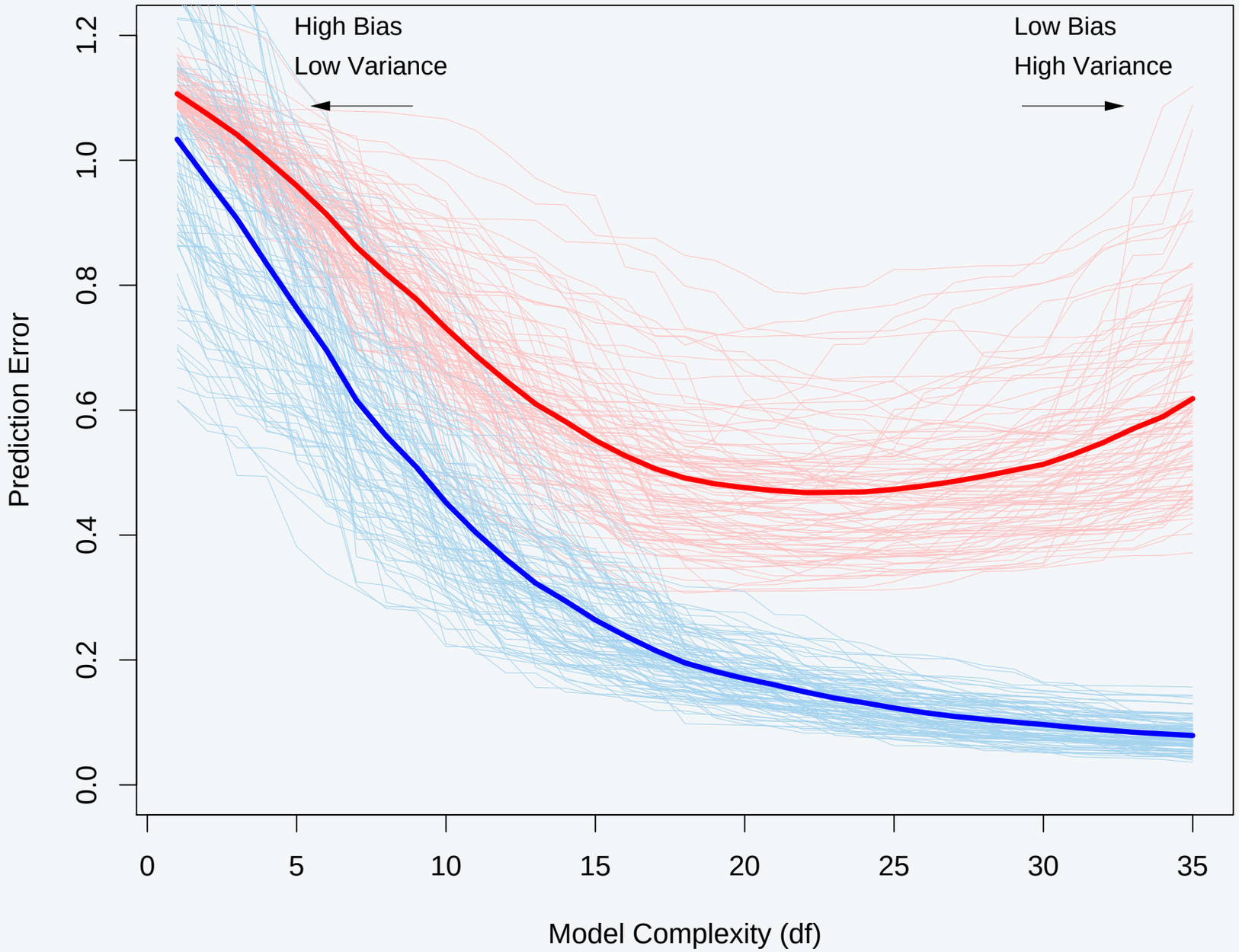
- An important issue to consider. There are always two competing forces that govern the choice of the best machine learning method, namely, bias and variance.

- Bias is the difference, on average, between the prediction and reality, across different possible samples of data.
  - The more flexible or complicated a method is, the more likely it will "get predictions right" (on average) across samples.
    - However, this comes at a cost.

- Variance refers to the amount that your prediction will vary across different possible samples of data.

- The more flexible or complicated a method is, the more likely it will "move around" depending on the sample.

Test and Training Error, and Model Complexity



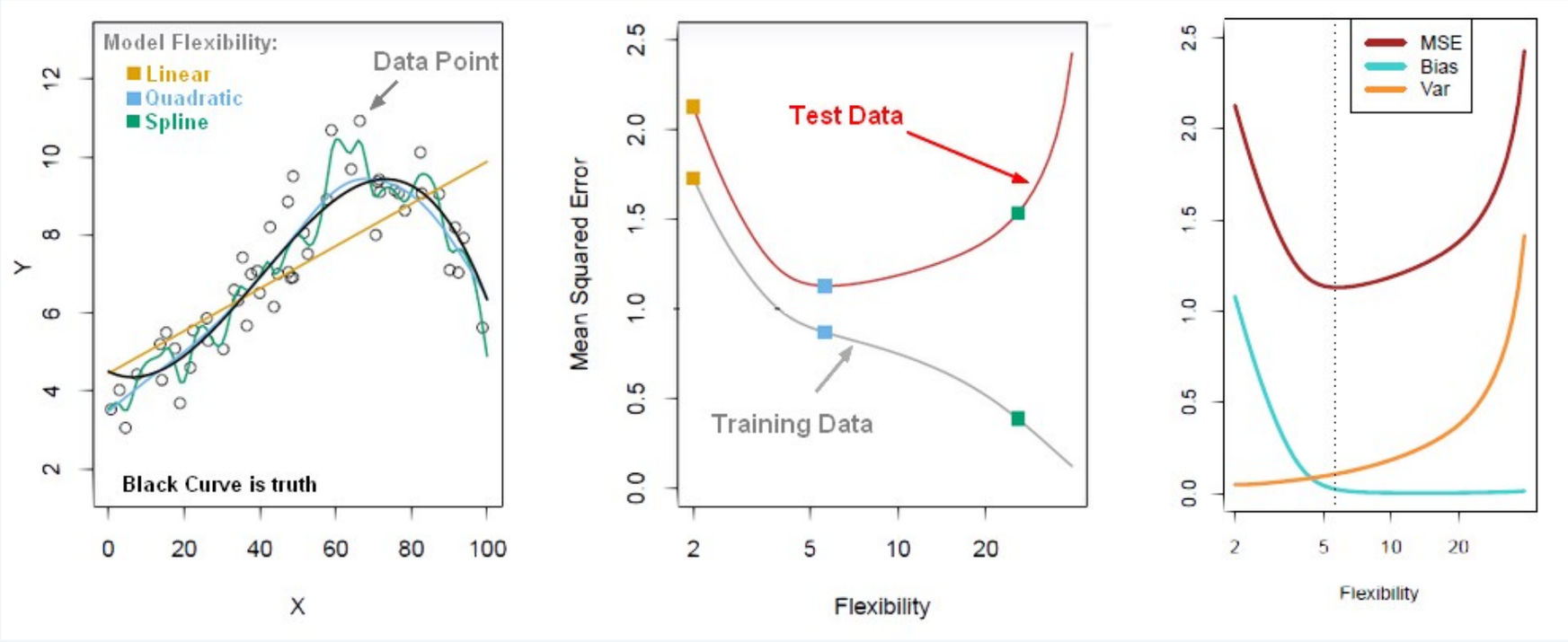
The Bias-Variance Tradeoff

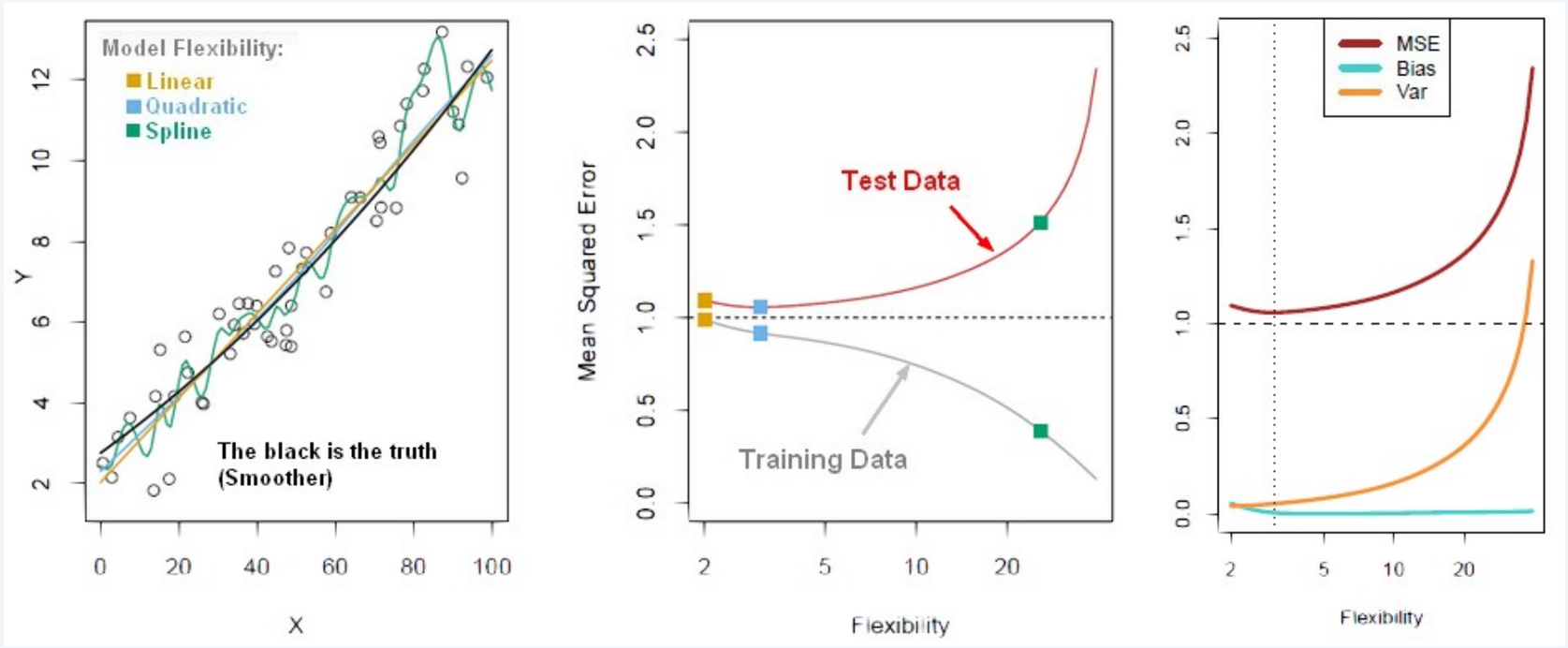
- It is possible to show that for any given $X = x$, the expected mean squared error (MSE) for a new Y generated for x will equal:

$$\begin{aligned} MSE &= \mathrm{E}\left[\left(Y - \hat{f}(x)\right)^2\right] \\ &= \mathrm{Bias}^2(\hat{f}(x)) + \mathrm{Variance}(\hat{f}(x)) + \sigma^2(\varepsilon) \end{aligned}$$

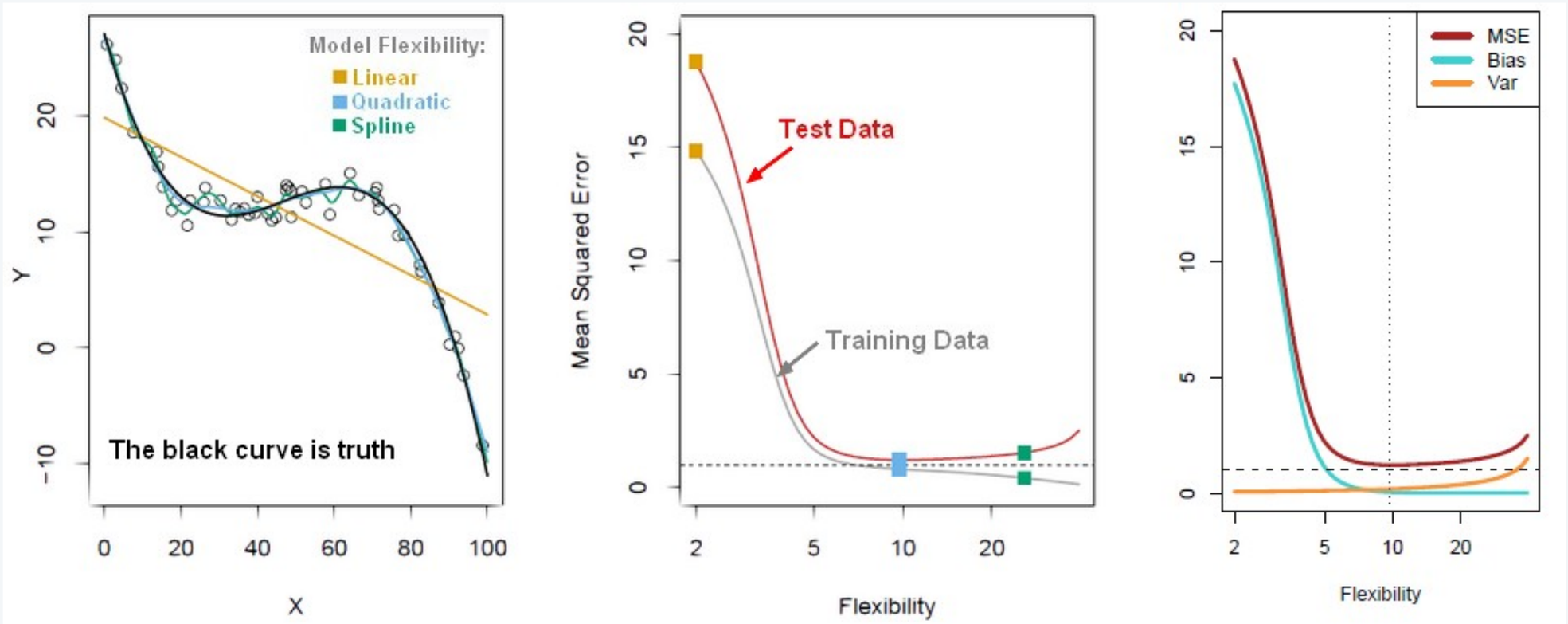
$$\begin{aligned} MSE &= \mathrm{E}\left[\left(Y - \hat{f}\right)^2\right] \\ &= \mathrm{E}\left[Y^2\right] + \mathrm{E}\left[\hat{f}^2\right] - 2\mathrm{E}\left[Y\hat{f}\right] \\ &= \mathrm{Var}\left[Y^2\right] + \mathrm{E}[Y]^2 + \mathrm{Var}\left[\hat{f}\right] + \mathrm{E}[\hat{f}]^2 - 2\mathrm{E}\left[(f+\varepsilon)\hat{f}\right] \\ &= \sigma^2(\varepsilon) + \mathrm{Var}\left[\hat{f}\right] + \left(f^2 + \mathrm{E}[\hat{f}]^2 - 2f\mathrm{E}\left[\hat{f}\right]\right) \\ &= \sigma^2(\varepsilon) + \mathrm{Var}\left[\hat{f}\right] + \left(f - \mathrm{E}[\hat{f}]\right)^2 \\ &= \sigma^2(\varepsilon) + \mathrm{Var}\left[\hat{f}\right] + \mathrm{Bias}^2 \end{aligned}$$

The Bias-Variance Tradeoff





The Bias-Variance Tradeoff



Selecting Training and Test Samples

- How do we select training and test samples (and predictors..)? Many possible approaches:
  - Split the data into fractions, train on a subset of them, test on the other subset.
  - "Pre-test" for particular predictors from a large set on the whole data, using univariate relationships, estimate a multivariate model using selected predictors.
- It will turn out that some methods are worse than others, and there are "rules of thumb" which are common in ML.
  - Standard approach is "k-fold cross-validation".
  - Also, any "pre-testing" or variable selection also needs to be done with great care.

K-Fold Cross Validation

- Basic principle of k-fold cross validation is to divide the available data into k equal parts (or "folds").

1	2	3	4	5
Train	Train	Validation	Train	Train

K-Fold Cross Validation

- The model is then fit on all of the training data, and the fitted model is then used to predict the response for the observations in the validation dataset.
  - The procedure is then repeated K times for different choices of validation dataset, and an average cross validation error is then computed.
- Mathematically, let $\kappa : \{1, ..., N\} \rightarrow \{1, ..., K\}$ map observations to folds, and let $\hat{f}^{-\kappa}(x, \alpha)$ represent the model (with tuning parameters α) fitted to the data with the κ'th fold removed. Then the cross-validation estimate of the prediction error is (L is loss function, explained later):

$$CV(\hat{f}, \alpha) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i, \alpha))$$

K-Fold Cross Validation

- Note: if there are specific tuning parameters α associated with model f, then we generally pick the parameters $\hat{\alpha}$ that minimize the CV-prediction error. Final model is $f(x, \hat{\alpha})$.

- As a general rule of thumb, most people pick $K = 5$ or 10.

  - If $K = N$, the resulting approach is called "leave one out cross-validation", or LOOCV.

  - Tradeoff between bias (low K) and variance (high K), also depends on sample size N.

  - Generally low K with low N will overestimate the prediction error rate.

Cross Validation the Wrong Way

- An approach that many people follow, which generates serious problems, is to "pre-select" predictors based on univariate correlations estimated on the entire sample.

- Typical strategy:

  - Find a subset of predictors that show strong univariate correlation with the outcomes.

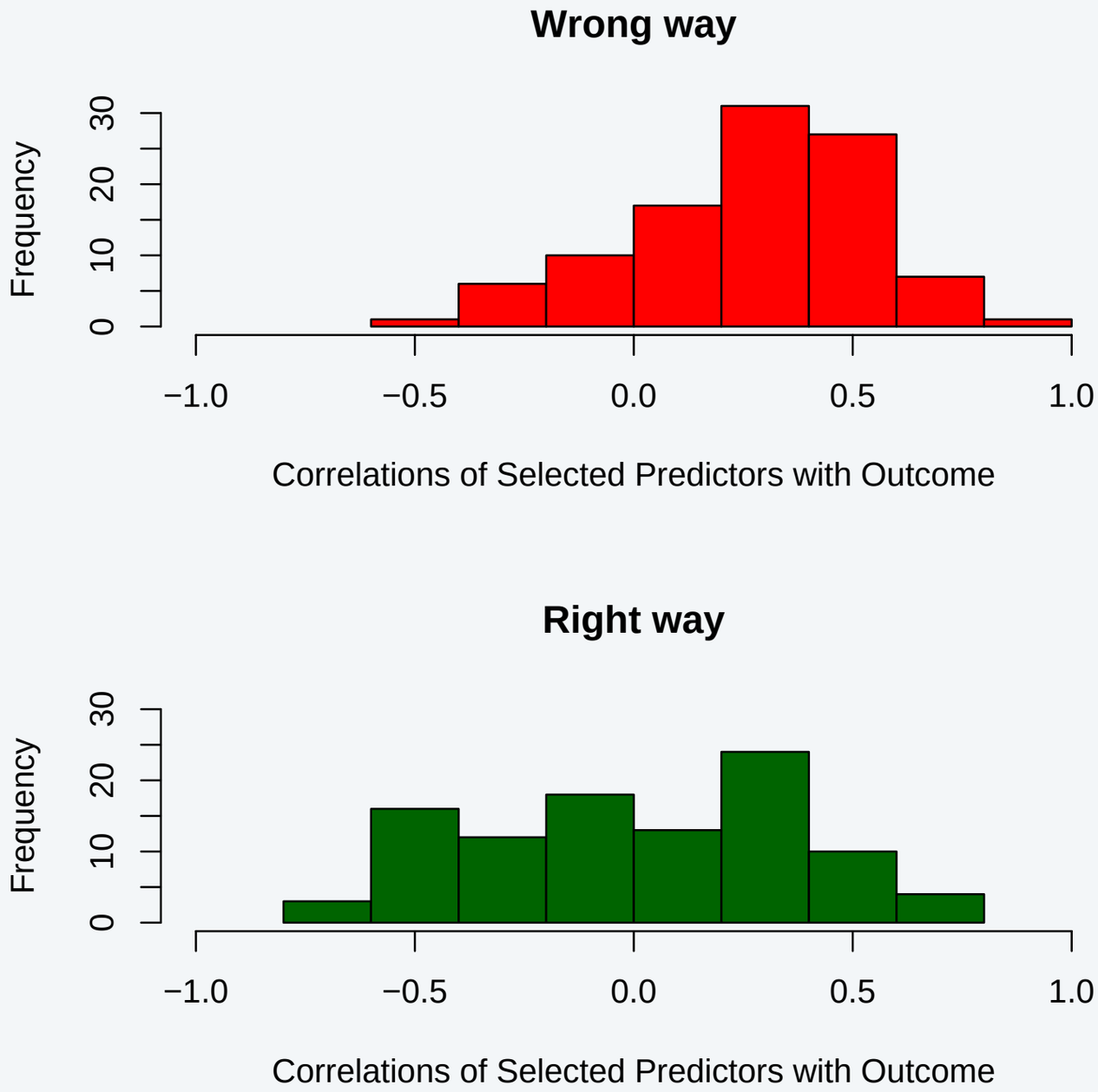
  - Build a multivariate classification or regression algorithm using this subset.

  - Use cross-validation to estimate tuning parameters and prediction error.

Cross Validation the Right Way

- Correct strategy:
  - First divide sample into K cross-validation folds.
  - For each fold κ, find a subset of good predictors using all of the other folds (the leave out κ sample).
  - Build the multivariate prediction model using these predictors and the leave out κ sample.
    - Evaluate the model on fold κ, repeat as normal. Note: selection of variables is also a tuning parameter in this setup.
- To see the difference between these two approaches, see an example next. The true model from which samples are simulated is one in which the outcomes are independent of the predictors.

Cross Validation the Right and Wrong Way



Classification: Concepts and Mathematical Foundations

- Consider the statistical learning model $Y = f(X) + \varepsilon$. To quantify prediction accuracy, we generally use a loss function. For example:
  - Squared loss: $L(Y, f(X)) = (f(X) - Y)^2$
  - Misclassification loss: $L(Y, f(X)) = 1$ if $f(X) \neq Y$; or 0 if $f(X) = Y$.

- Suppose we observe samples $(x_1, y_1), ..., (x_n, y_n)$ from the joint distribution of (X, Y), then the expected loss $\mathrm{E}_{(X,Y)}[L(Y, f(X)]$ can be estimated as:

$$\hat{\mathrm{E}}_{(X,Y)}[L(Y, f(X)] = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))$$

Bayes Classifier

- Now, a classifier f^* that minimizes the expected loss for a classification problem is called the Bayes classifier (or is termed Bayes optimal).
- To find it, we can solve:

$$f^* = \arg\min_f \mathrm{E}_{(X,Y)}[L(Y, f(X)] .$$
$$= \arg\min_f \int \mathrm{E}_Y[L(Y, f(X)|X = x]g(x)dx$$

Note: this need not be unique.

The Bayes Optimal Classifier

- Let us abuse notation and say that $Y = k$ denotes when Y belongs to the k^{th} class.

- Now assume a simple $0-1$ loss function. Then, the expected loss at any $X = x$ can be written as:

$$\begin{aligned}
\mathrm{E}_Y[L(Y, f(X)|x] &= \sum_{k=1}^{K} L(k, f(X))P(Y = k|X = x) \\
&= \sum_{k=1}^{K} P(Y \neq f(X)|X = x) \\
&= 1 - P(Y = f(X)|X = x)
\end{aligned}$$

- Put differently, the Bayes optimal classifier should be theoretically easy to find – simply pick the class with the higest posterior probability – this is given by Bayes rule (we will see this).

The Bayes Optimal Classifier

- What is this saying in words?
- Suppose I pick a classifier f. Then the expected loss of f evaluated at a specific value of $X = x$ can be computed easily.
- First, go through each of the classes. For each class, compute the true average number of times that $Y = k$ when $X = x$. Then, use these average probabilities to weight the losses in each case.
  - When Y is truly k and the classifier delivers a different answer, you are penalized by 1, otherwise there is no penalty.
- This specific form of the loss function delivers a nice result. For each class, check the average number of times the classifier incorrectly says Y belongs to it. Simply sum this across classes!
  - But this is just $1-$ the average number of times the classifier gets it right (i.e., says $Y = k$ when this is the truth).

The Bayes Optimal Classifier

- Back to finding the Bayes optimal classifier. We now know we need to pick the class to solve:

$$f^*_{Bayes} = \arg \max_{k=1,\ldots,K} P(Y = k|X = x)$$

- But from Bayes' theorem,

$$P(Y = k|X = x) = \frac{P(X = x|Y = k)P(Y = k)}{P(X = x)}$$

The Bayes Optimal Classifier

- Now, note that for a (k-class) classification problem, we can write $P(X = x)$ as:

$$g(x) = \sum_{k=1}^{K} \pi_k g_k(x),$$

where $P(Y = k) = \pi_k$ is the probability of class k, and $g_k(x)$ is the conditional density of X given $Y = k$, i.e., $P(X = x|Y = k)P(Y = k) = \pi_k g_k(x)$.

- So, $f^*_{Bayes} = \arg\max_{k=1,\ldots,K} \frac{\pi_k g_k(x)}{\sum_{k=1}^{K} \pi_k g_k(x)}$, i.e., the Bayes optimal classifier assigns the class for which the posterior probability is the highest.

Why all the Fuss?

- The Bayes classifier is a very useful theoretical benchmark, as it is the best possible classifier.
- However, it is clearly impossible to implement, since in reality, we don't know the true values of π_k (i.e., the true probability that $Y = k$), or of $g_k(x)$, i.e., the conditional density of of X given $Y = k$.
- So we need to develop estimators of these quantities.
- The entire classification problem can be boiled down to a search for good estimators of $\pi_k g_k(x), k = 1, ..., K$. Note: this is called the discriminant function.

Why all the Fuss?

- The Bayes classifier is a very useful theoretical benchmark, as it is the best possible classifier.
- However, it is clearly impossible to implement, since in reality, we don't know the true values of π_k (i.e., the true probability that $Y = k$), or of $g_k(x)$, i.e., the conditional density of of X given $Y = k$.
- So we need to develop estimators of these quantities.
- The entire classification problem can be boiled down to a search for good estimators of $\pi_k g_k(x), k = 1, ..., K$. Note: this is called the discriminant function.
- Understanding Bayes optimality is a good way to fix these ideas.

Logistic Models

- What we are after is an estimate of $P(Y = k|X = x)$. To explain how logistic regression fits in, let's think of a simply binary classification problem, i.e., one in which the outcome variable is either 0 or 1 (think default forecasting, for example).

- A simple way to think of the decision is:

$$Y = 1 \text{ if } \log \frac{P(Y = 1|X = x)}{P(Y = 0|X = x)} > 0, \text{ and } Y = 0 \text{ otherwise.}$$

Logistic Models

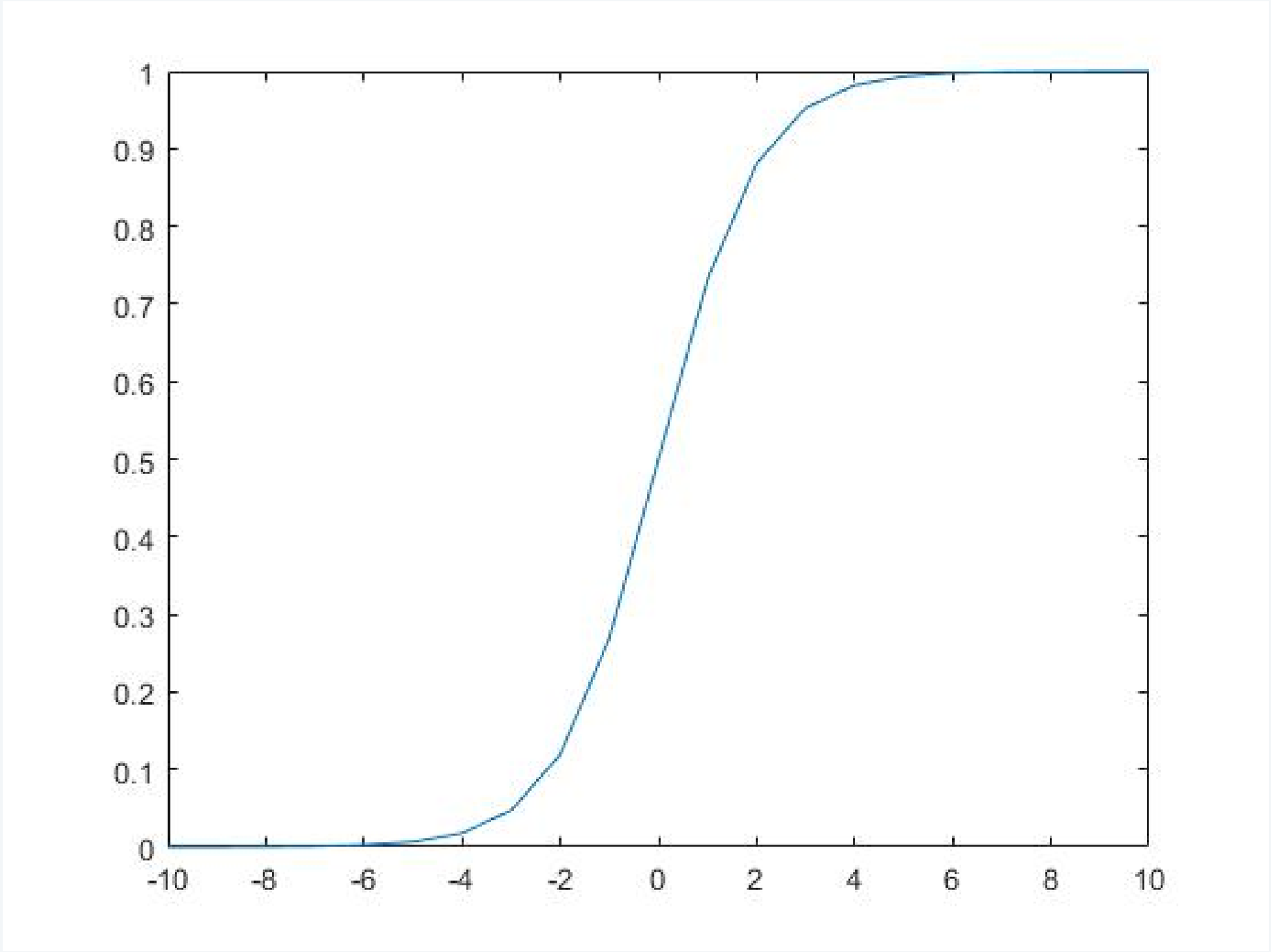
- Of course, we don't know $P(Y|X)$, but suppose we set the log odds ratio described above to have a linear form, i.e.,

$$\log \frac{P(Y=1|X=x)}{1-P(Y=1|X=x)} = f(X,\beta) = \beta_0 + \beta_1 X,$$

- And clearly, this models conditional probabilities as (easily generalizable for multiple X variables):

$$P(Y=1|X=x) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}; P(Y=0|X=x) = \frac{1}{1+e^{\beta_0+\beta_1 X}}.$$

The Logit Function



Logit Classification

- Once the logit is estimated (using standard maximum likelihood), the prediction that minimizes the misclassification rate is $Y = 1$ when $P(Y = 1|X = x) \geq 0.5$ and $Y = 0$ otherwise.

- This is equivalent to guessing $Y = 1$ when $\beta_0 + \beta_1 X \geq 0$ and 0 otherwise, in other words, logit provides a linear classifier.
  - The decision boundary if X is one dimensional is a point on the line.
  - The decision boundary if X is two-dimensional is a line, and so on...

- Note that the use of the logistic function also imposes some strong assumptions.
  - The probability of being in a particular class depends on the distance from the boundary, with movement towards the extremes being quicker when X and β are large.

Linear Discriminant Analysis

- Back to the Bayes classifier:

$$f^*_{Bayes} = \arg \max_{k=1,\ldots,K} \frac{\pi_k g_k(x)}{\sum_{k=1}^{K} \pi_k g_k(x)}$$

How can we implement this? If we had π_k, $g_k(x)$, we could make headway. Since we don't, need to come up with estimates of these quantities.

- Linear discriminant analysis makes the following assumptions:
  - $\hat{\pi}_k$ (the estimated probability that Y belongs to the kth class) is just the proportion of training observations that belong to the kth class.
  - $g_k(x)$ is Gaussian, with mean μ_k, variance σ^2, which doesn't vary across classes k.

Linear Discriminant Analysis

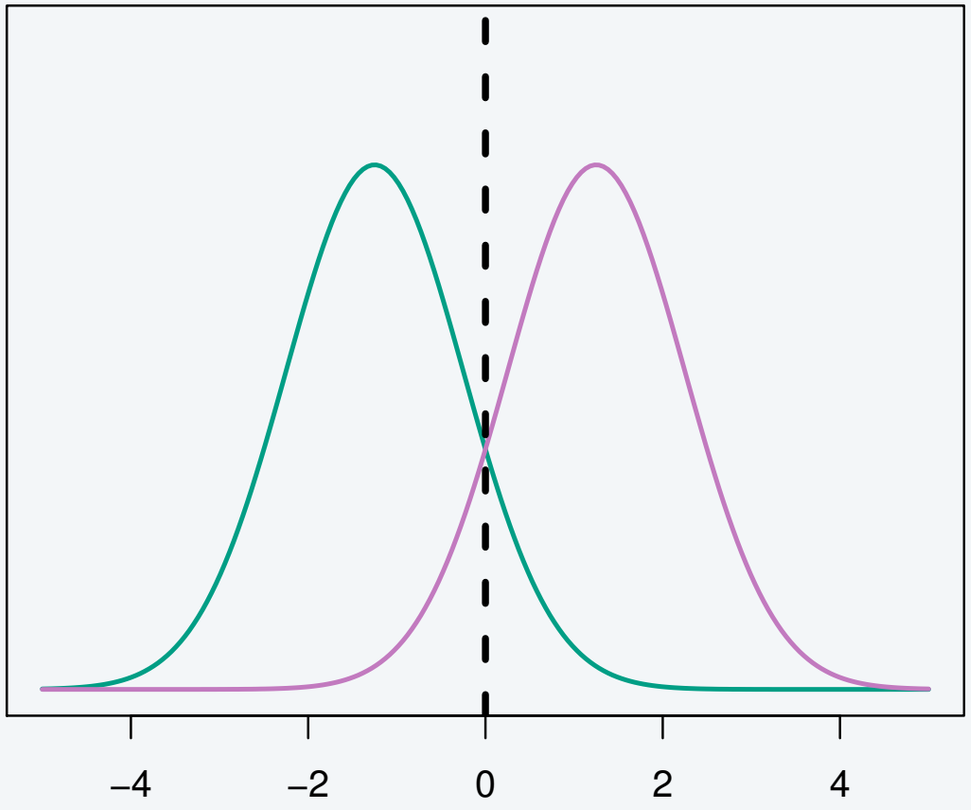
- To estimate these quantities for LDA we simply use (for univariate X, with analogues for the multivariate version):

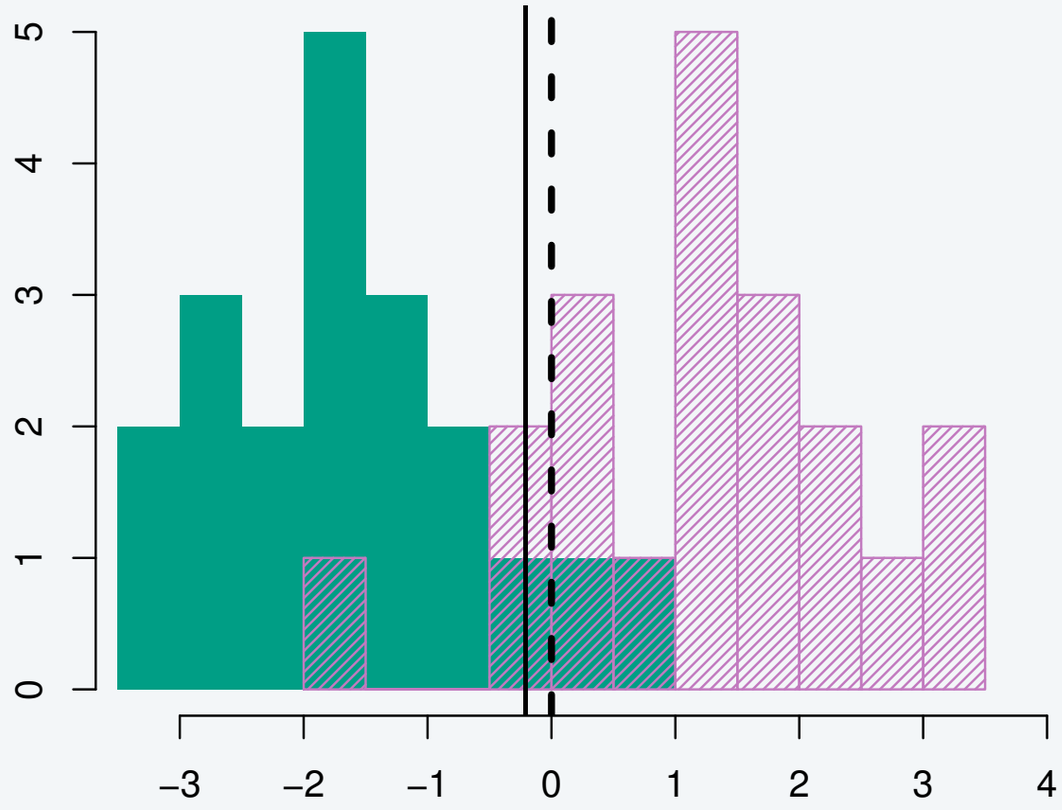
$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

Linear Discriminant Analysis

Two class example, equiprobable case.





Linear and Quadratic Discriminant Analysis

(Please see ISL pp. 140-144)

- Why is it known as Linear discriminant analysis? For multivariate X, the classifier assigns an observation $X = x$ to the class for which:

$$\delta_k(x) = x^{'}\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^{'}\Sigma^{-1}\mu_k + \log \pi_k$$

  is greatest, which is simply a linear function of x.

- Quadratic discriminant analysis simply modifies this by relaxing the assumption that the covariance matrices are the same across the k classes. For multivariate X, the QDA classifier assigns an observation $X = x$ to the class for which:

$$\delta_k(x) = x^{'}\Sigma_k^{-1}\mu_k - \frac{1}{2}\mu_k^{'}\Sigma_k^{-1}\mu_k + \log \pi_k - \frac{1}{2}\log |\Sigma_k| - \frac{1}{2}x^{'}\Sigma_k^{-1}x,$$

  which, with the addition of the two extra terms, is now a quadratic function of x.

Linear and Quadratic Discriminant Analysis

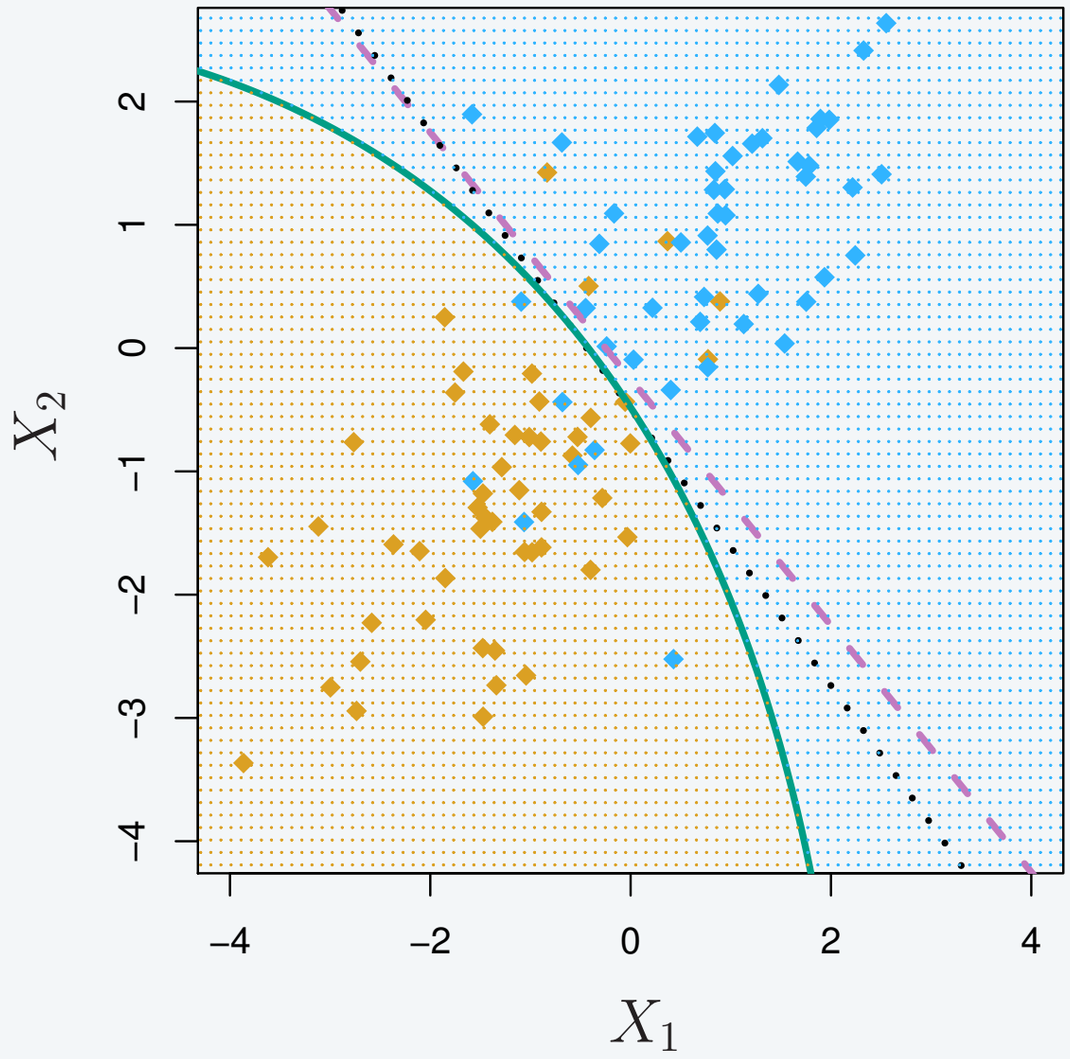
- Which one is better? Depends - there is a bias-variance tradeoff as always.

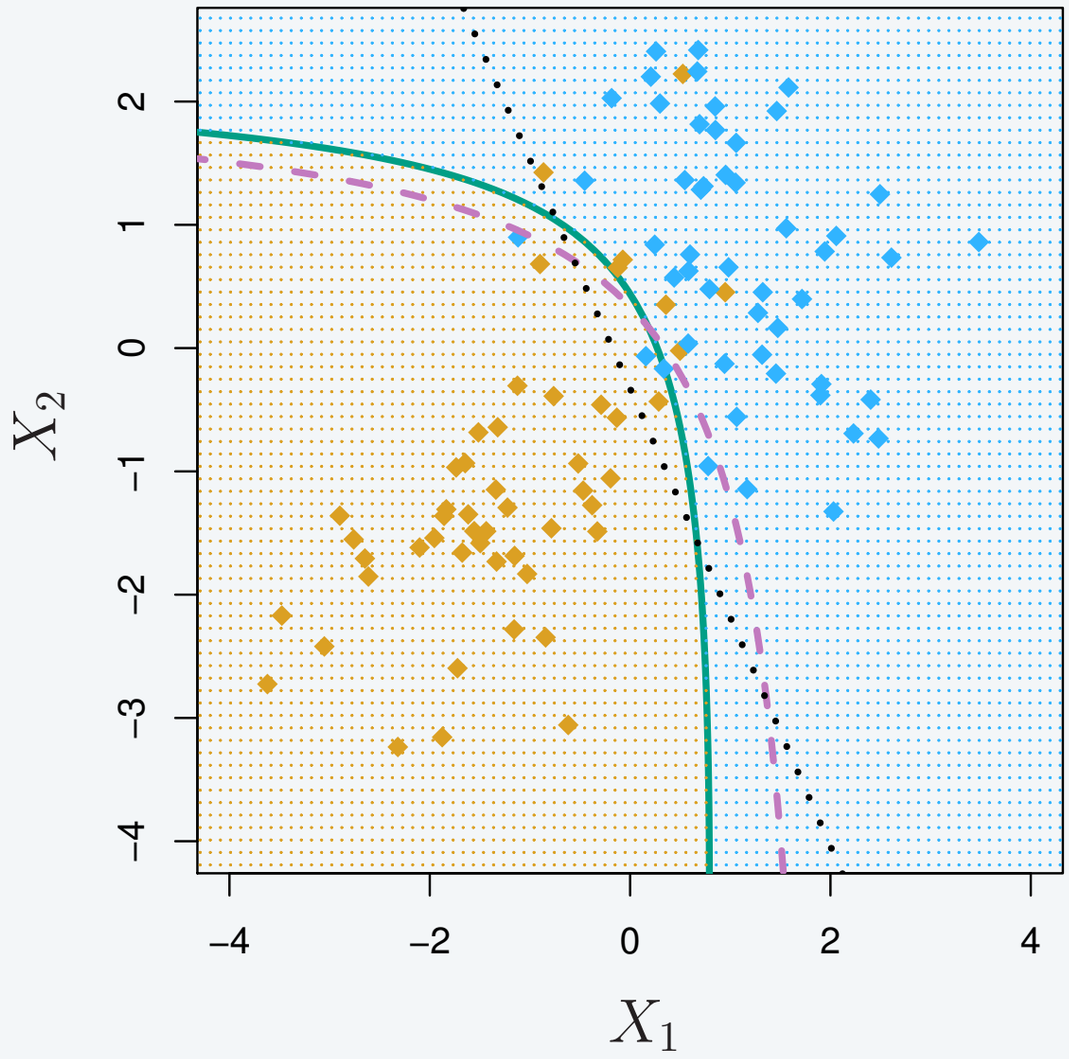
- LDA has fewer parameters to estimate, since the additional requirement to estimate a covariance matrix per class is onerous.

- QDA relaxes this assumption, and may therefore be more accurate, but the high number of additional predictors generates variance.

Linear and Quadratic Discriminant Analysis

Purple: Bayes, Black: LDA, Green: QDA





Non-Parametric Approaches

- The logistic, LDA, and QDA approaches all impose specific parametric assumptions on the data.
  - LDA and QDA assume that $g_k(x)$ is Gaussian.
  - The logit approach assumes that the log odds ratio is linear in X.

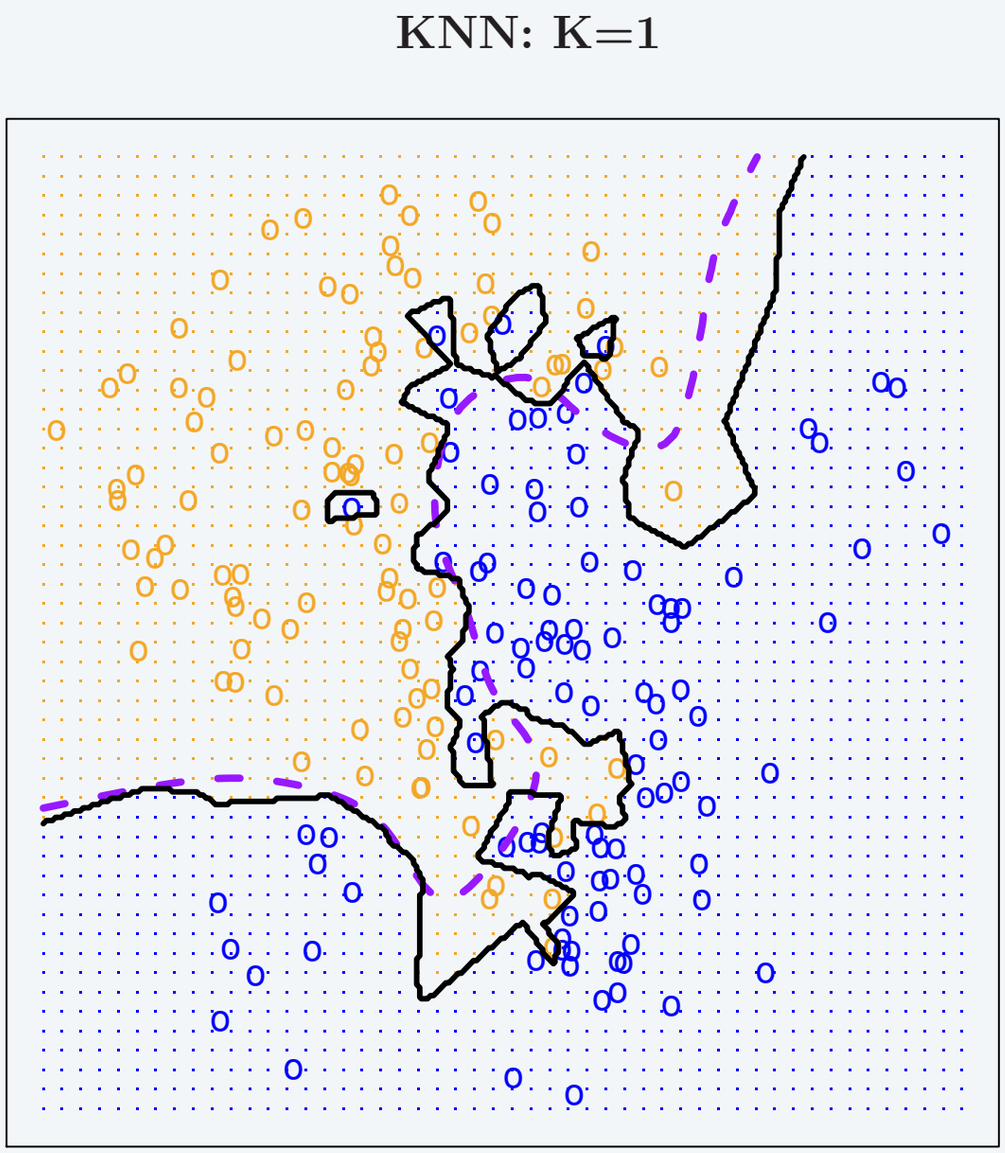
- Two approaches that impose fewer parametric assumptions on the data are K-Nearest Neighbours (KNN), and Trees.

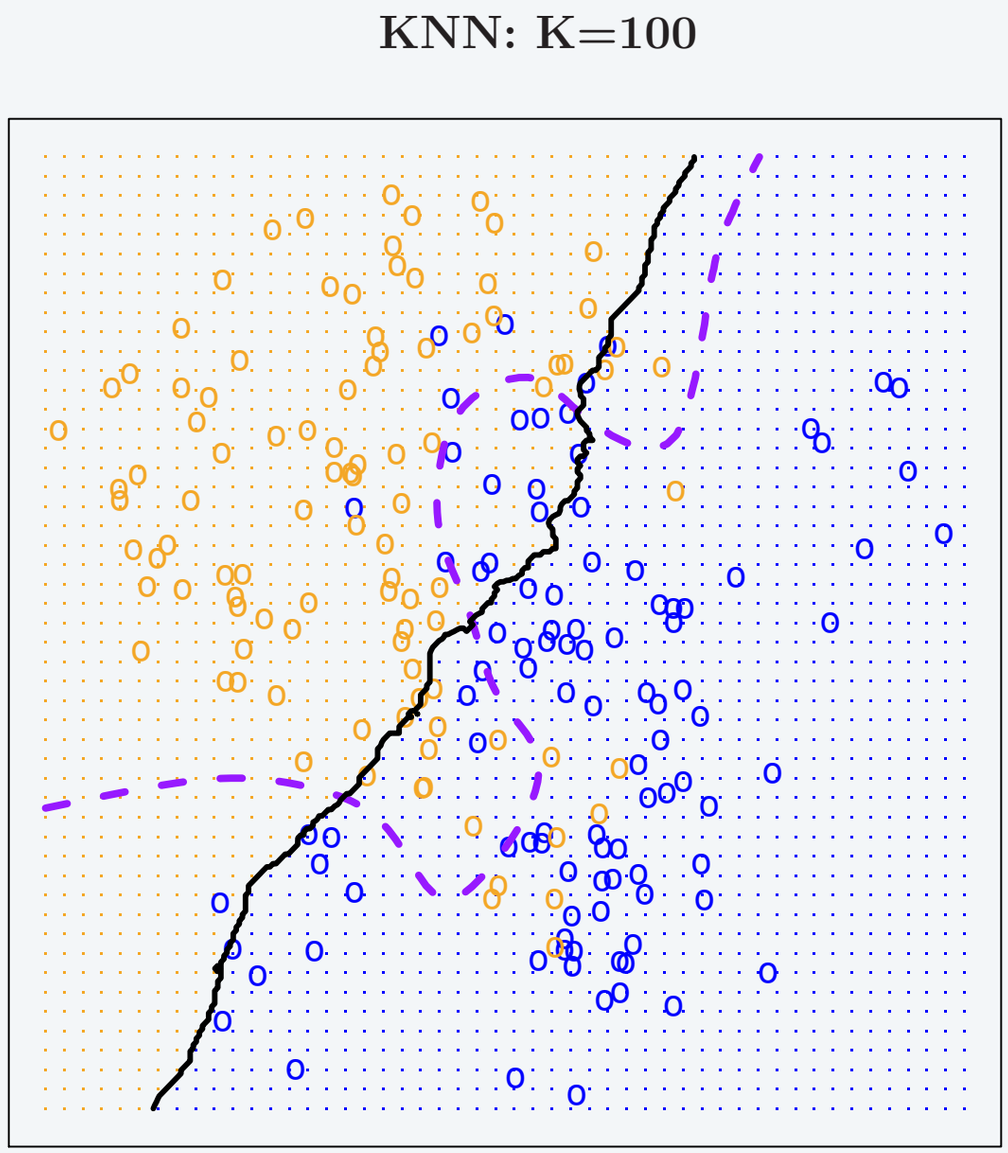
- KNN is simple to explain. For any new $X = x$, it finds the K points in the training data that are closest to x (we can denote this set as $\mathbb{N}_0$), and then simply estimates the conditional probability of $Y = k$ for each class using the fraction of points in $\mathbb{N}_0$ that are in the class.

$$P(Y = k|X = x) = \frac{1}{K} \sum_{i \in \mathbb{N}_0} I(y_i = k).$$

An Illustration of KNN with Different Choices of K

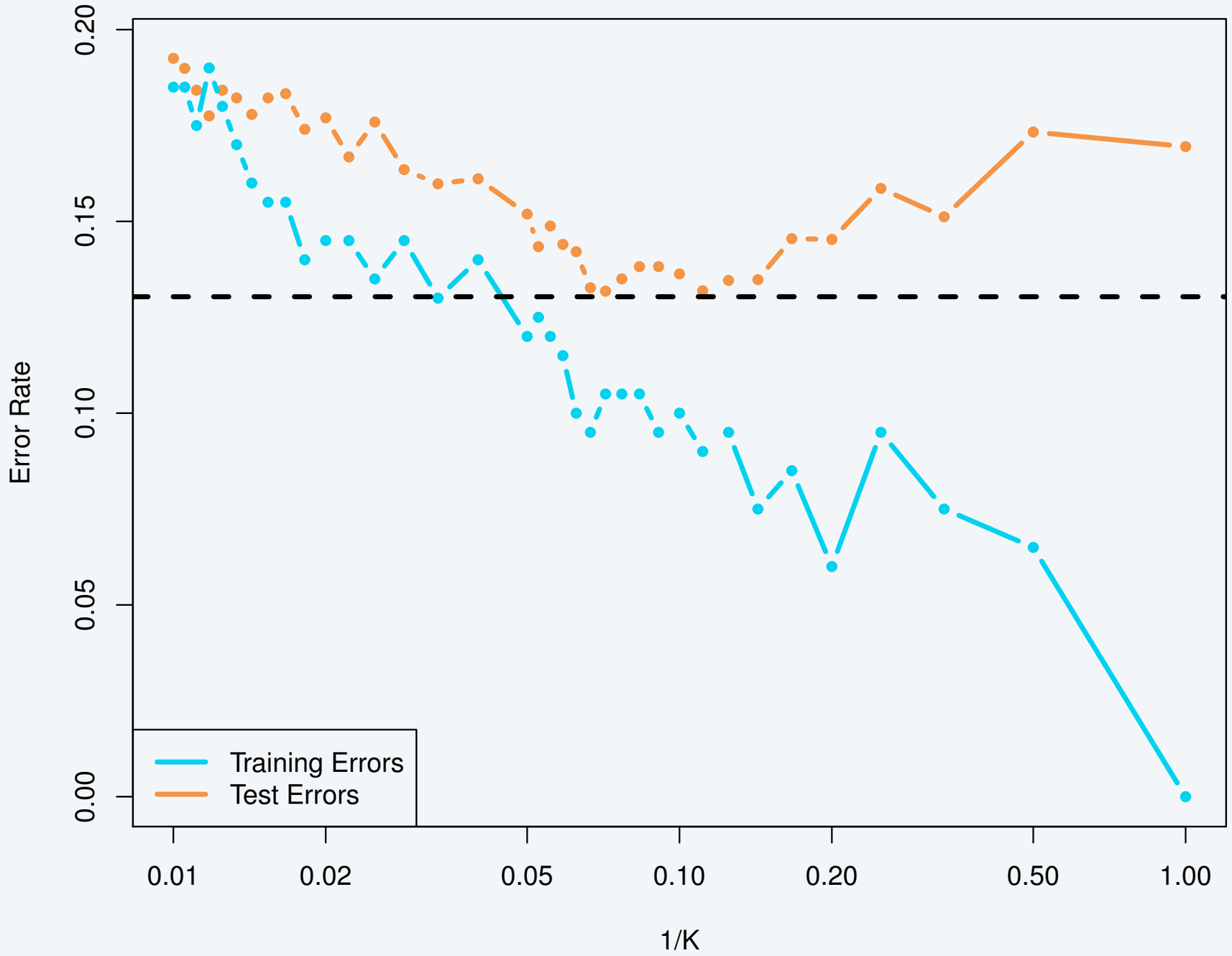
Purple: Bayes, Black: KNN





Training and Test Error Rates with Different

K



Classification Trees

- The basic idea of trees is to partition the space of X into subspaces, and then to estimate the classification outcome for new units as the most commonly occurring class in the training set with values in the same subspace.

- The partitioning is sequential - called "recursive binary splitting," one covariate at a time.

- Guidance for making these splits is provided by some criterion.

Classification Trees: Splitting

- One way to split is to compute the classification error rate as the fraction of training observations in each region that does not belong to the most commonly occurring class, i.e., $1 - \max_k \hat{p}_{mk}$ where $\hat{p}_{mk}$ is the proportion of training observations in the mth region belonging to the kth class.

- Another (preferable) one is the Gini index: $G = \Sigma_k \hat{p}_{mk}(1 - \hat{p}_{mk})$. Clearly, this will be low when all $\hat{p}_{mk}$ observations are close to 0 or 1, i.e., nodes are "pure".

- A third is called cross-entropy: $D = -\Sigma_k \hat{p}_{mk} \log \hat{p}_{mk}$, which has similar properties to the Gini index.

A Tree Cookbook - I

- Pick any X variable X_p. Pick a threshold t and consider splitting the data on this threshold, i.e., depending on whether

$$X_{i,p} \leq t \text{ versus } X_{i,p} > t$$

- Let the estimator then be:

$$f_{p,t}(x) = \left\{ \begin{array}{l} \max_k \hat{p}_k(X_{i,p} \leq t) \text{ for } x_p \leq t \\ \max_k \hat{p}_k(X_{i,p} > t) \text{ for } x_p > t \end{array} \right\}$$

- Then compute the criterion function (say it's the Gini G), and find both the covariate p^* and the threshold t^* that solves:

$$\arg \min_{p,t} G(f_{p,t}(x))$$

A Tree Cookbook - II

- Partition the covariate space into two subspaces based on whether $X_{i,p^*} \leq t^*$ or not.

- Repeat, splitting each subspace in the way that leads to the biggest improvement in the objective function.

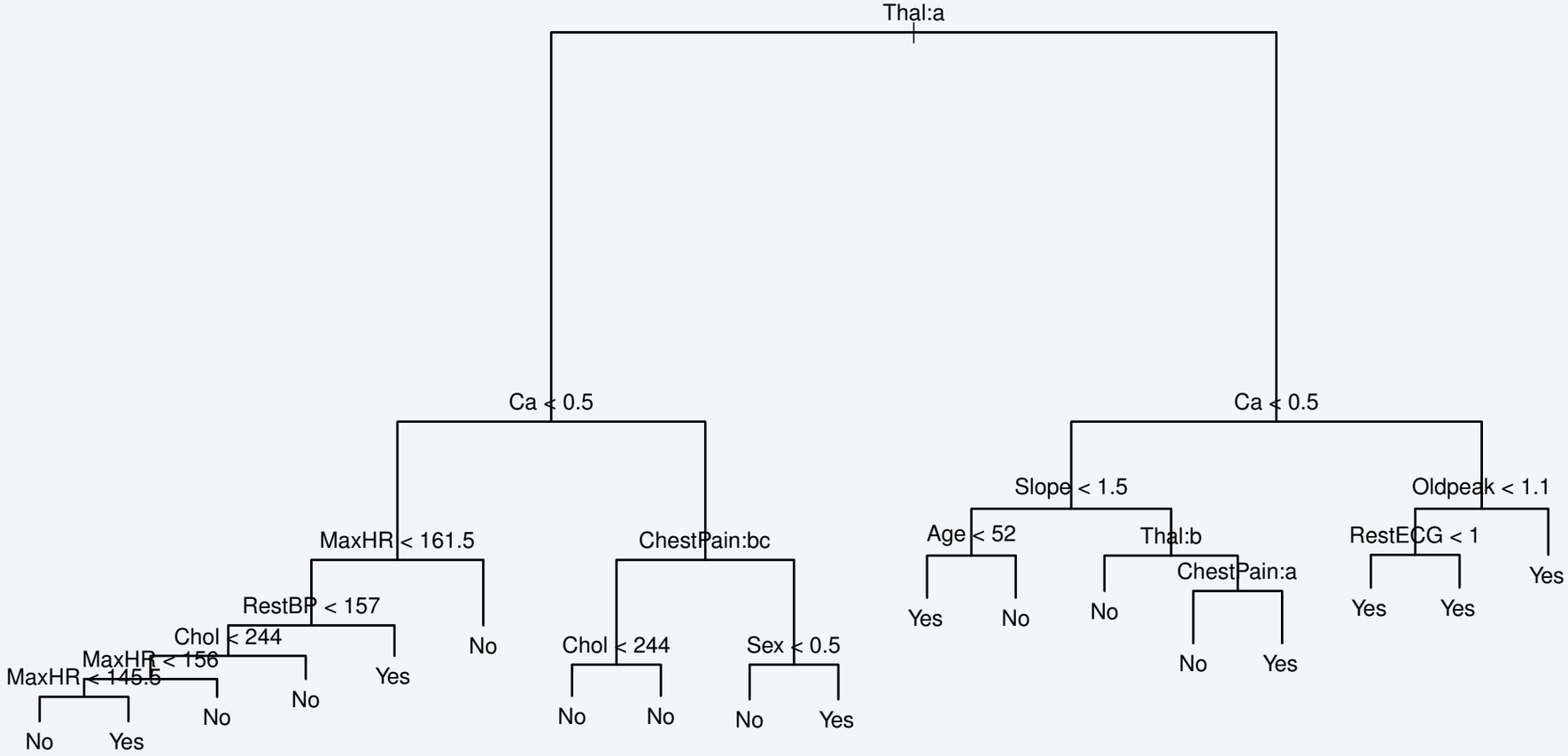
- Keep splitting the subspaces to minimize the objective function, optionally, with a penalty λ for the number of splits (also called leaves).

- - That is, with penalty, objective function is now $Q = G(f_{p,t}(x)) + \lambda(\#\text{ of leaves})$.

Pruning the Tree

- A frequently used approach is to grow a big tree by using a deliberately small value of the penalty term, or simply growing the tree till the leaves have a preset small number of observations.

- Then go back and prune branches or leaves that do not collectively improve the objective function sufficiently.



Conclusion and Next Week

- This has been a whirlwind tour through the broad contour of machine learning approaches, and a number of useful classification approaches.

- Next week, we will look at applying some of these techniques (along with a series of associated challenges) to credit default forecasting.

- We will begin by learning some of the theoretical finance foundations behind credit default. Then the guest lecturer will help us contrast a number of basic ML approaches on real data, and then

- Note: forecasting on its own will simply not be enough...It's the combination of finance insight and ML techniques that holds great promise for the future.

Imperial College
Business School

Imperial means
Intelligent Business