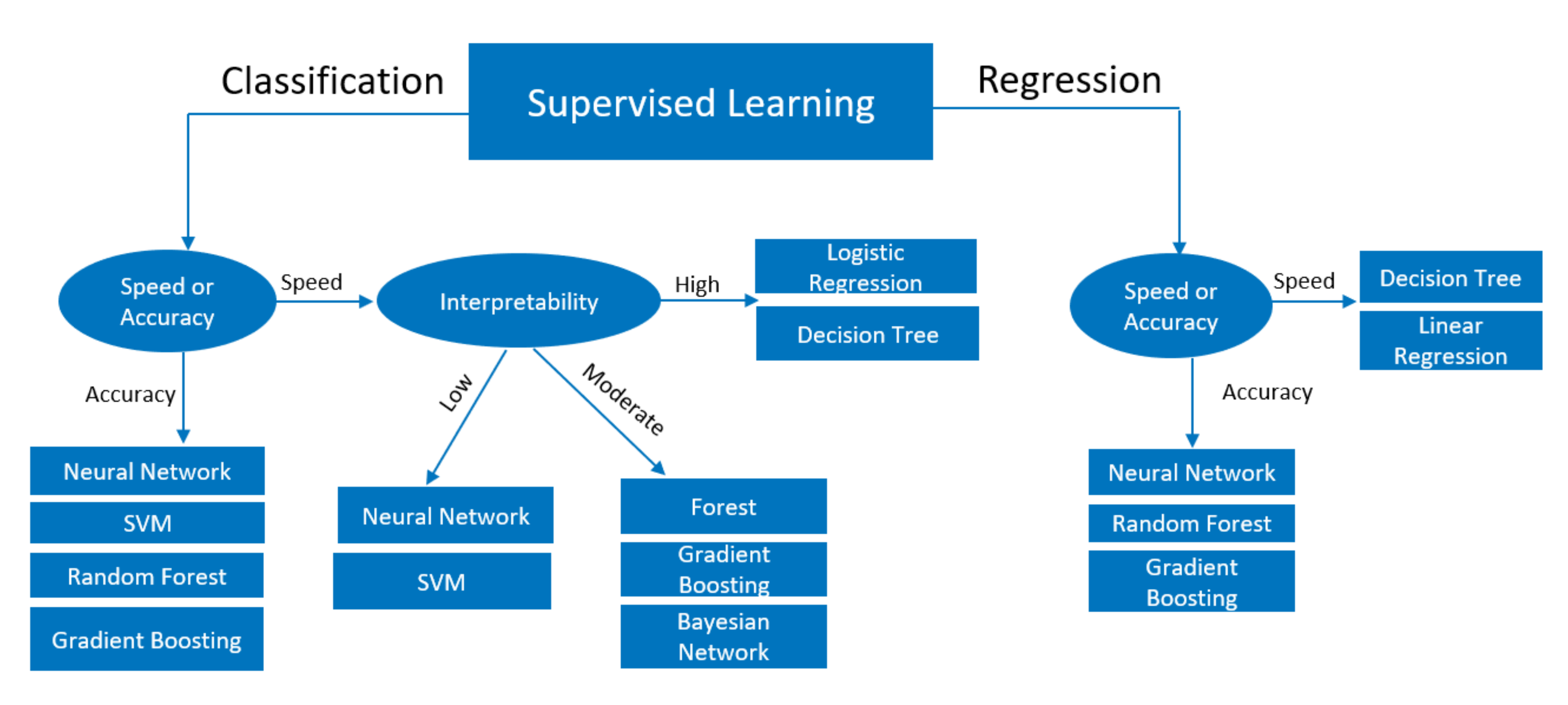
Wrap-Up Topics and Final Exam Prep

Agenda

- Support Vector Machines – Quick Intro
- Neural Networks – Quick Intro
- “Wrap-up” topics
- Final exam logistics
- Final exam topics and sample questions

Modeling Algorithm Selection Guidelines



Introduction to Support Vector Machines

Support Vector Machines

Overview

- Comes out of the computer science field (not statistics)
- Considered to be one of the best ways of doing classification
- Highly flexible – automatically determines any relationship between predictors and response
  - Don't need to specify relationship before modeling
- Tend to be "black boxes"
- Used in fields such as image classification, handwriting recognition, financial decision making, and text mining

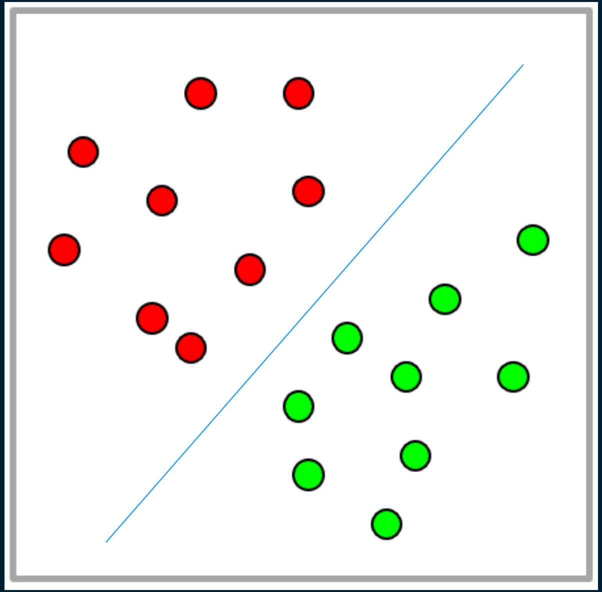
Support Vector Machines

Overview

- Approaches two-class classification problems directly:
  - Attempt to find a plane that separates the classes in a feature space
- If we cannot find such a plane, two techniques are used:
  - Soften what we mean by “separates”
  - Enrich and enlarge the feature space so that separation is possible

Support Vector Machines

- Originally designed solely for “decisions” (classification)
- Have been expanded to also provide ranks and estimates

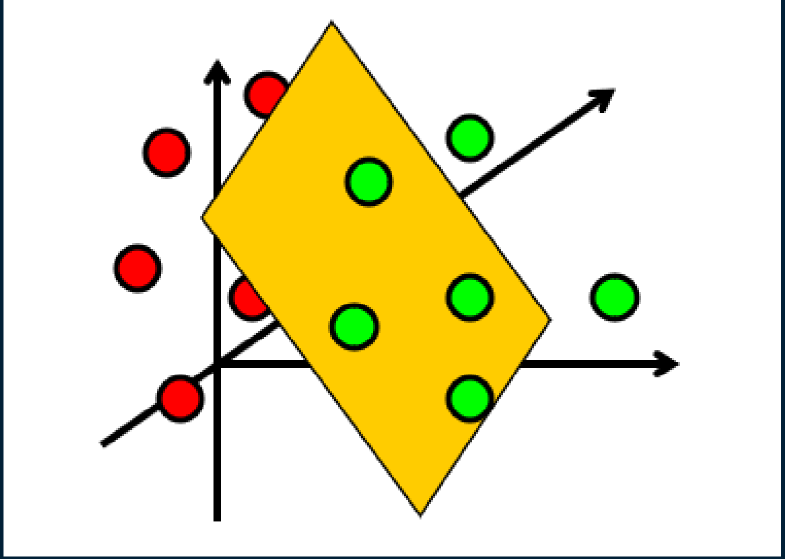


- Line is the support vector “machine”
- Also called “classifier”, “classifier model”, or a “classification rule”

Binary classification model with two-dimensional input space

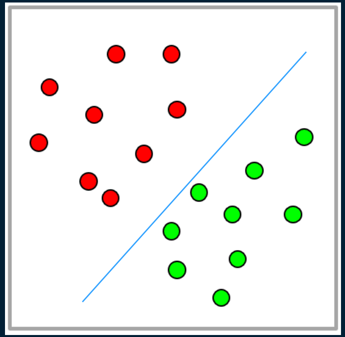
Support Vector Machines

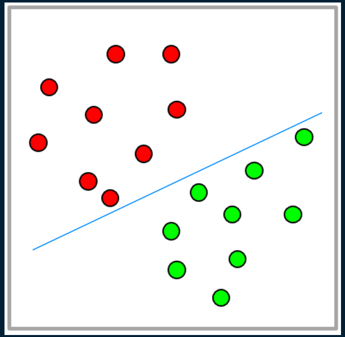
Planes

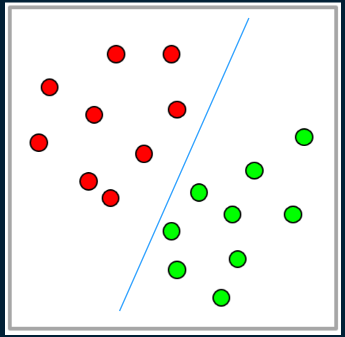


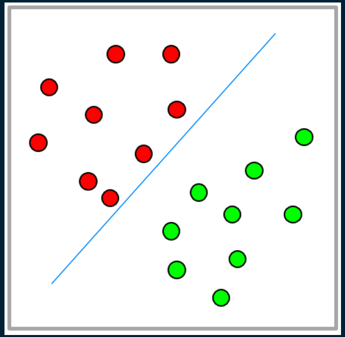
Binary classification model with three-dimensional input space

Which Hyperplane is the Best One?



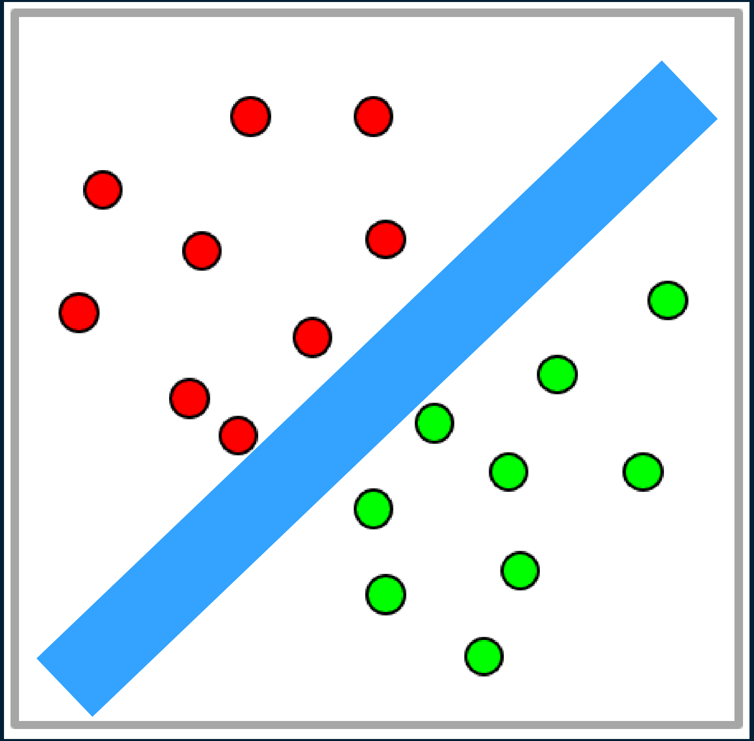






Which Hyperplane is Best?

A “Fat” Hyperplane

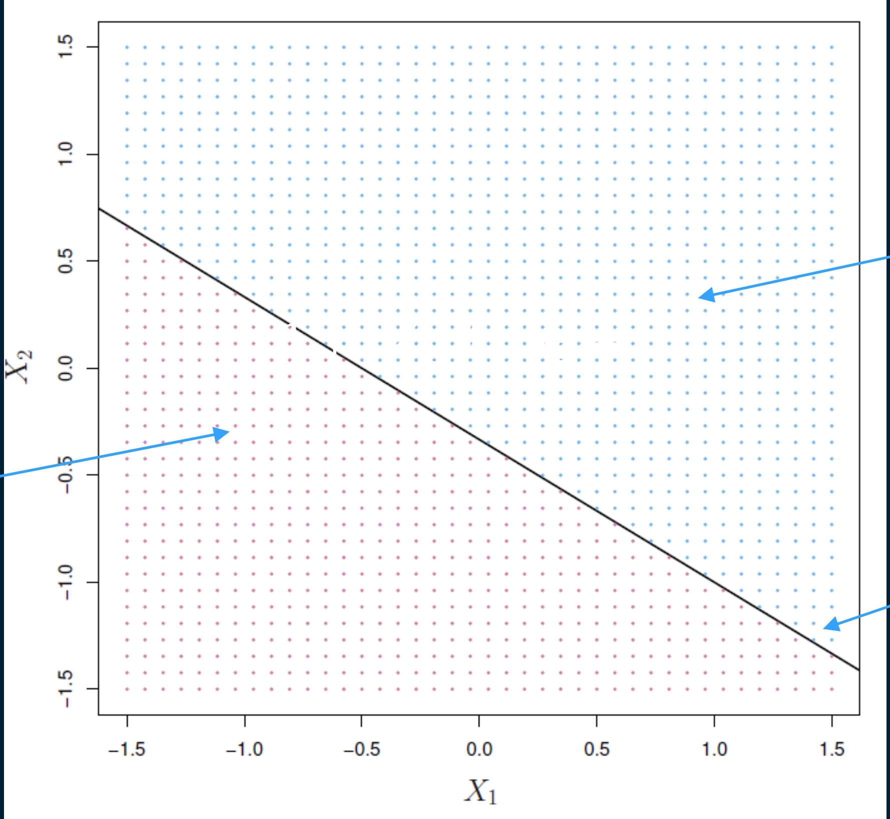


Hyperplane Definition

- A hyperplane in p dimensions is a flat subspace of dimension $p-1$
  - If $p=2$, hyperplane is a line
  - If $p=3$, hyperplane is a traditional plane
- Hyperplane equation:

$$\beta_0+\beta_1X_1+\beta_2X_2+\ \ldots+\beta_pX_p=0$$

Hyperplane



Region of points with
$1 + 2X_1 + 3X_2 > 0$

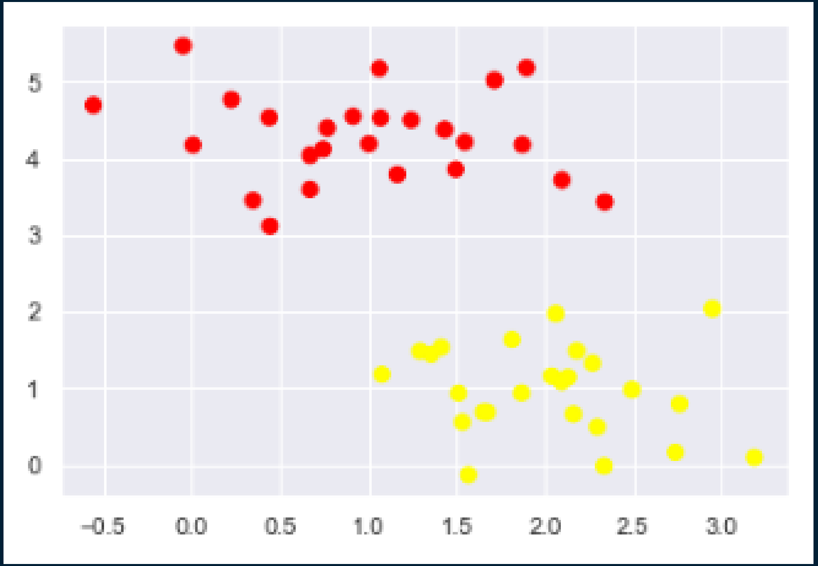
Region of points with
$1 + 2X_1 + 3X_2 < 0$

$$1 + 2X_1 + 3X_2 = 0$$

Separating Hyperplanes

Simple Example

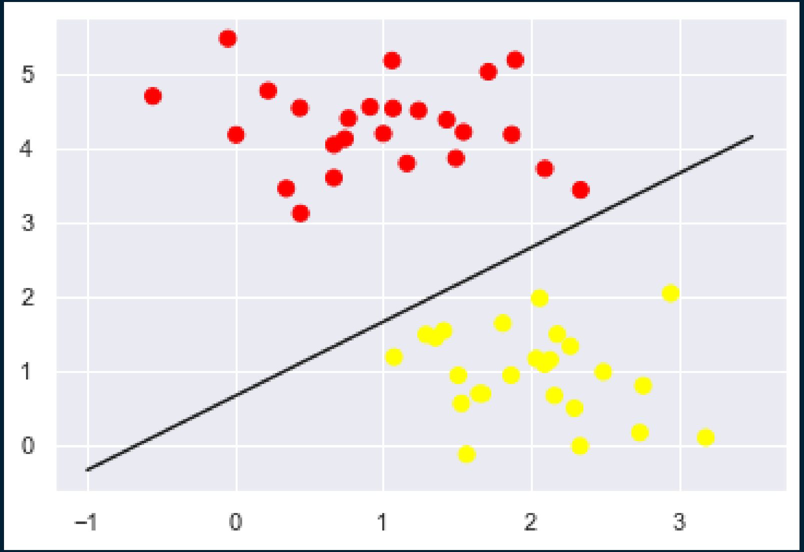
Two clearly-separate sets



Separating Hyperplanes

Simple Example

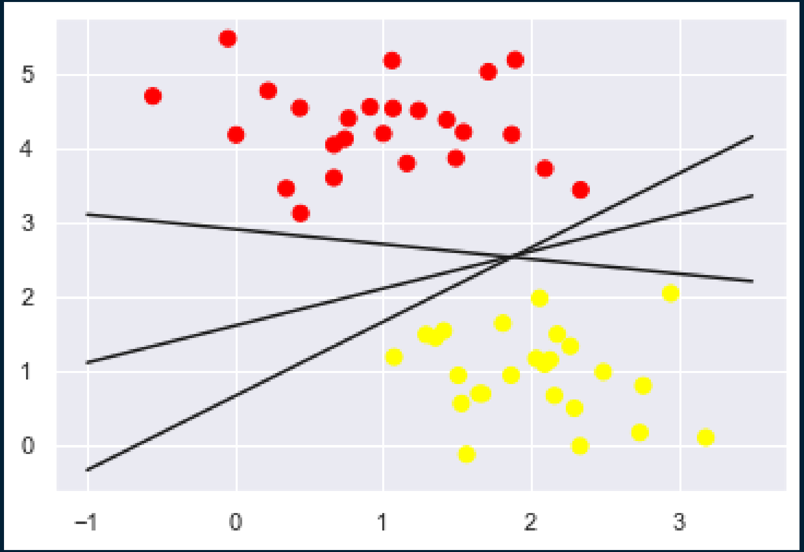
Easy to find a line
that separates them



Separating Hyperplanes

Simple Example

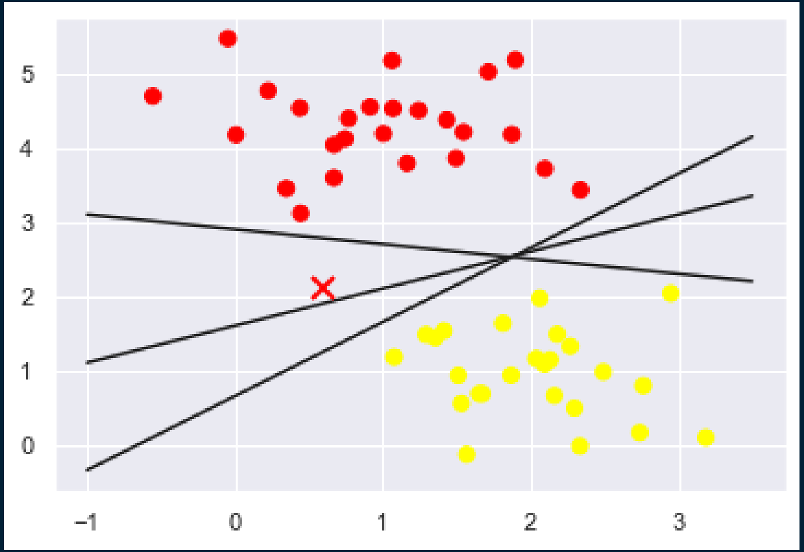
Actually, many!



Separating Hyperplanes

Simple Example

Which can lead to confusion when predicting classifications for new observations – like the “X” here

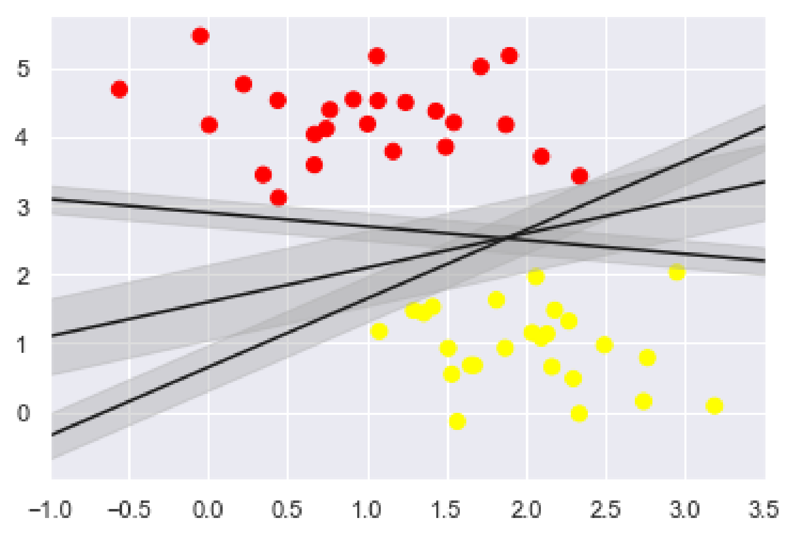


Separating Hyperplanes

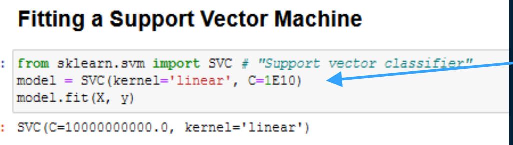
Maximizing the Margin

Basic approach:

- Draw a margin around each line that goes up to the nearest point
- Select the line that has the largest margin



Fitting a Support Vector Machine

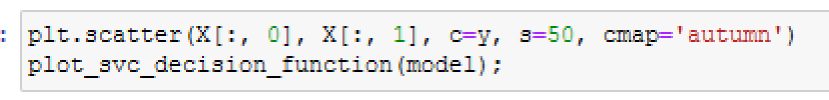


Fitting a Support Vector Machine

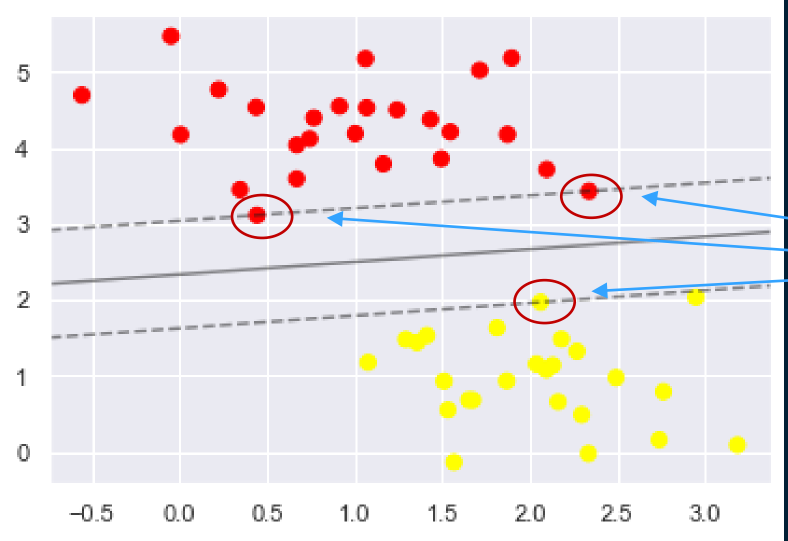
```
from sklearn.svm import SVC # "Support vector classifier"
model = SVC(kernel='linear', C=1E10)
model.fit(X, y)
```

```
SVC(C=10000000000.0, kernel='linear')
```

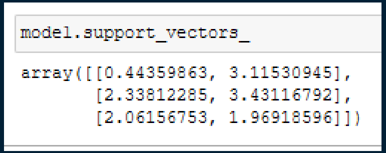
Note the two model parameters. We will re-visit shortly.



```
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn')
plot_svc_decision_function(model);
```



“Support Vectors”

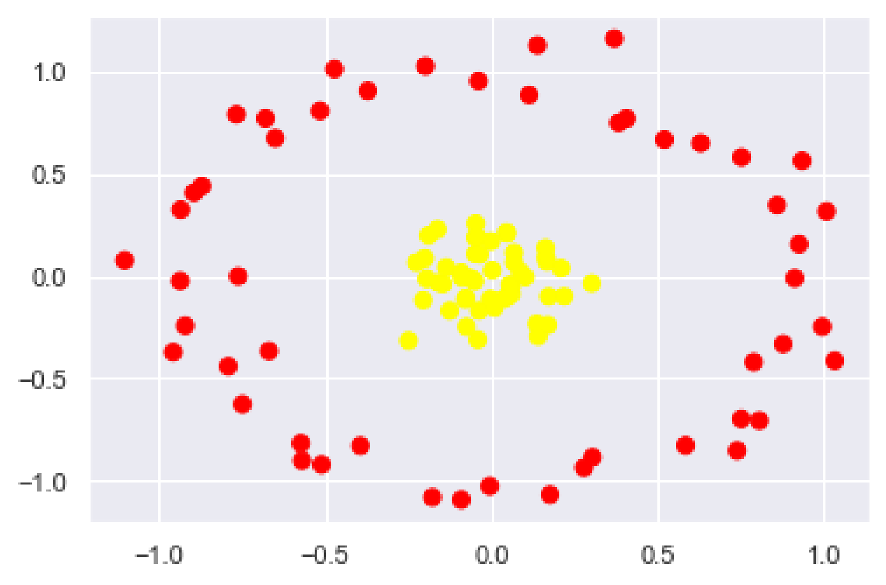


```
model.support_vectors_
```

```
array([[0.44359863, 3.11530945],
       [2.33812285, 3.43116792],
       [2.06156753, 1.96918596]])
```

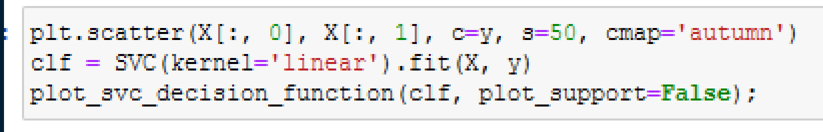
Beyond Linear Boundaries

Clearly Non-Separable Classes

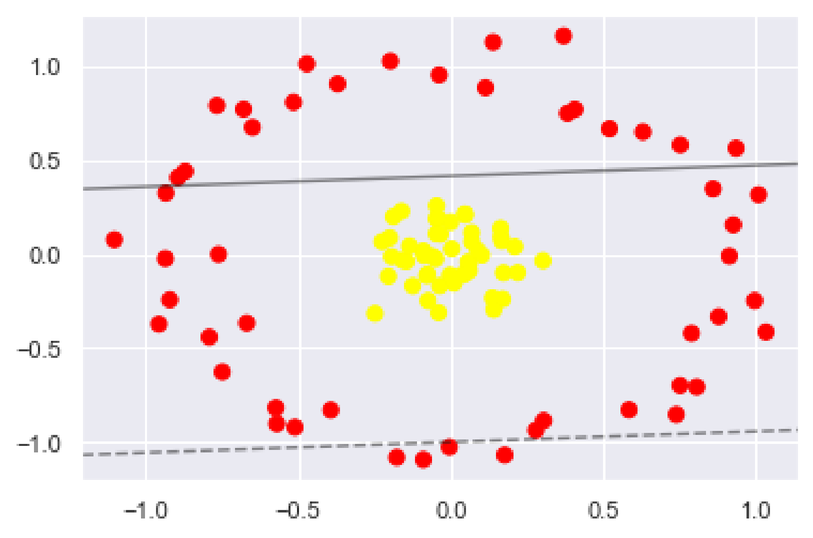


Beyond Linear Boundaries

Clearly Non-Separable Classes (Linearly)



```
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn')
clf = SVC(kernel='linear').fit(X, y)
plot_svc_decision_function(clf, plot_support=False);
```



Beyond Linear Boundaries

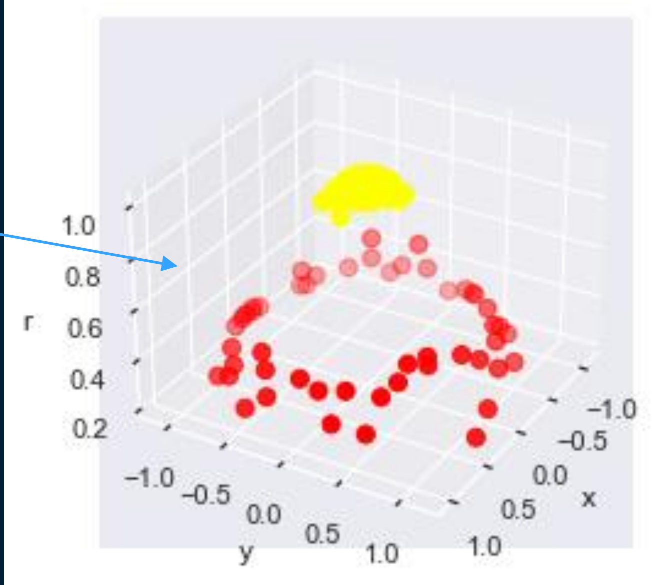
- SVMs become powerful when combined with kernels
  - Projections onto a higher dimension (e.g., polynomial expansions) can enable linear separators to be drawn in this higher dimension
- One popular kernel type in SVMs is radial basis functions (RBF)
- For this example, we try a simple projection:
  - $r_i = \sum_{j=1}^{2} e^{-X_{i,j}^2}$

Will have largest values for X values closest to 0

Beyond Linear Boundaries

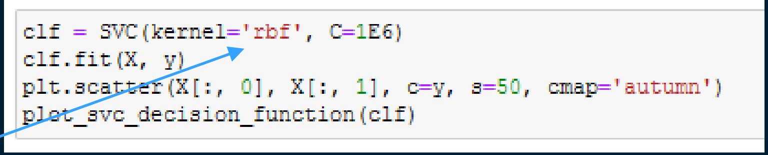
Adding RBF Centered on 0

It’s apparent that a 2-D plane could now separate the classes

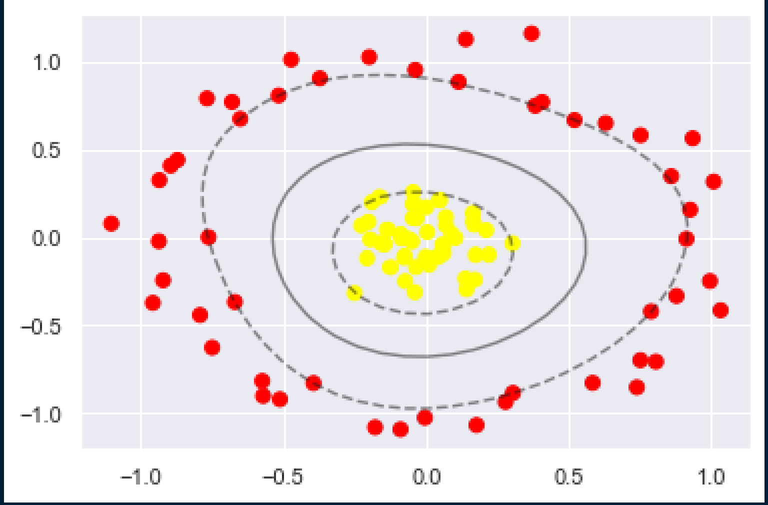


Beyond Linear Boundaries

Adding RBF Centered on 0



```
clf = SVC(kernel='rbf', C=1E6)
clf.fit(X, y)
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn')
plot_svc_decision_function(clf)
```



Software computes basis function centers at every point in dataset and identifies ideal function

Beyond Linear Boundaries

Softening Margins

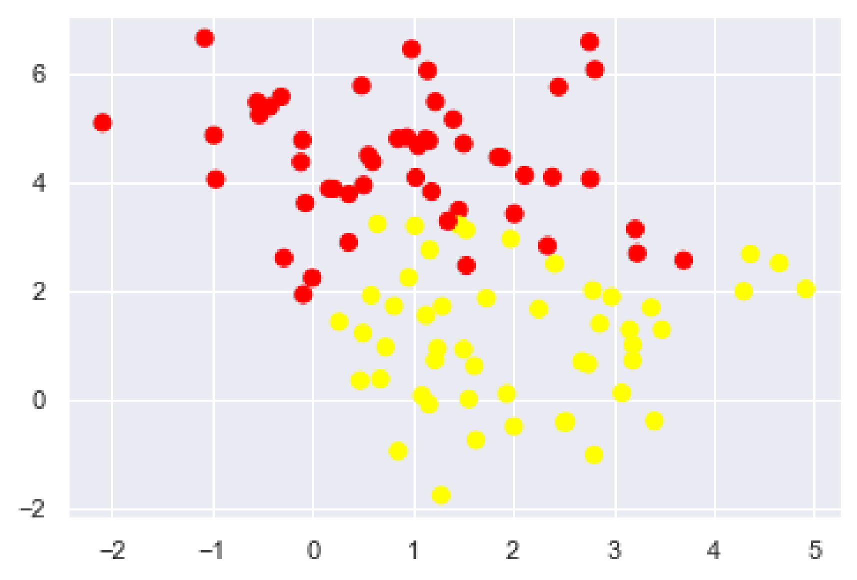
- However, this basis function “trick” only works when a perfect decision boundary exists, but what if your data has some overlap with every possible basis expansion?

Beyond Linear Boundaries

Softening Margins

There may be no cases where the boundary can be perfectly separated, so we allow some number of points on the other side of the margin.

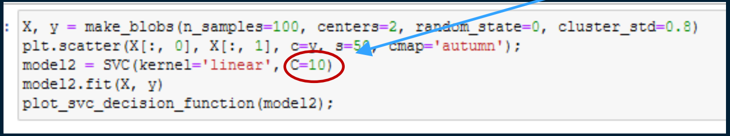
The "hardness" of the margin is controlled by the tuning parameter "C"



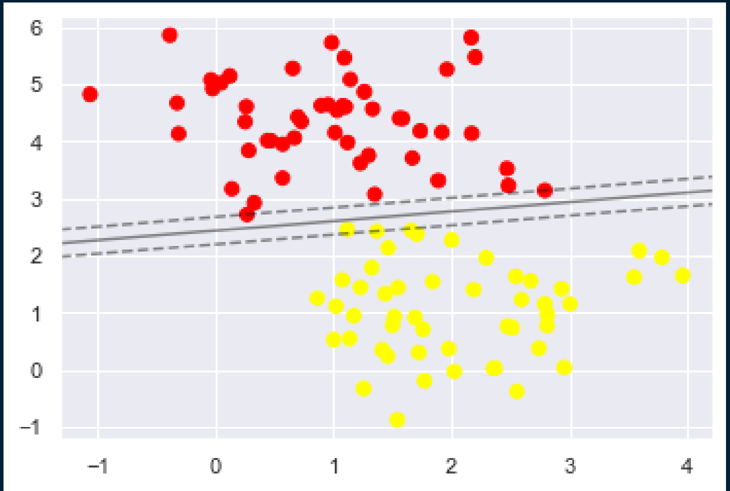
Beyond Linear Boundaries

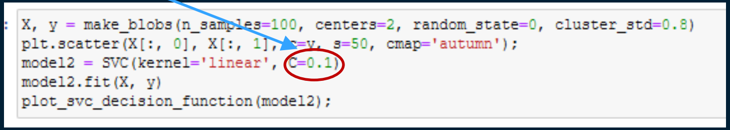
Softening Margins

Margin “softness” controlled by C parameter

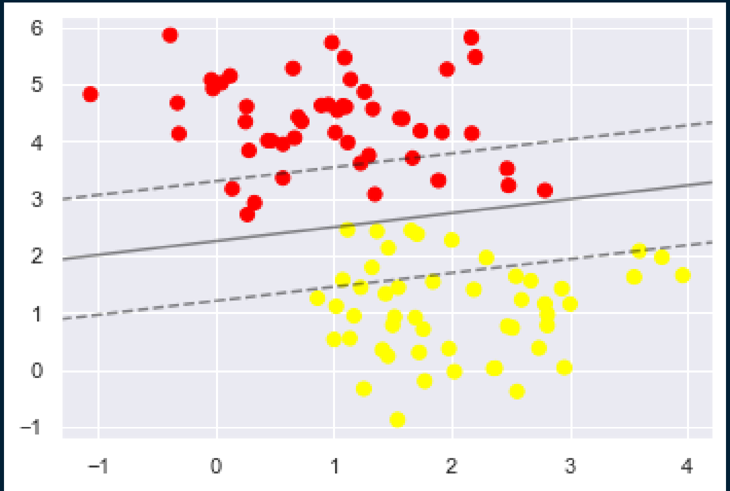


```
X, y = make_blobs(n_samples=100, centers=2, random_state=0, cluster_std=0.8)
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn');
model2 = SVC(kernel='linear', C=10)
model2.fit(X, y)
plot_svc_decision_function(model2);
```





```
X, y = make_blobs(n_samples=100, centers=2, random_state=0, cluster_std=0.8)
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn');
model2 = SVC(kernel='linear', C=0.1)
model2.fit(X, y)
plot_svc_decision_function(model2);
```



Parameters for SVMs for Classification

- The penalty C (regularization term)
- The kernel function and its parameters

Summary of SVM

An *SVM* is a hyperplane with a maximum-margin in a feature space, constructed by use of a kernel function in the input space.

Advantages:

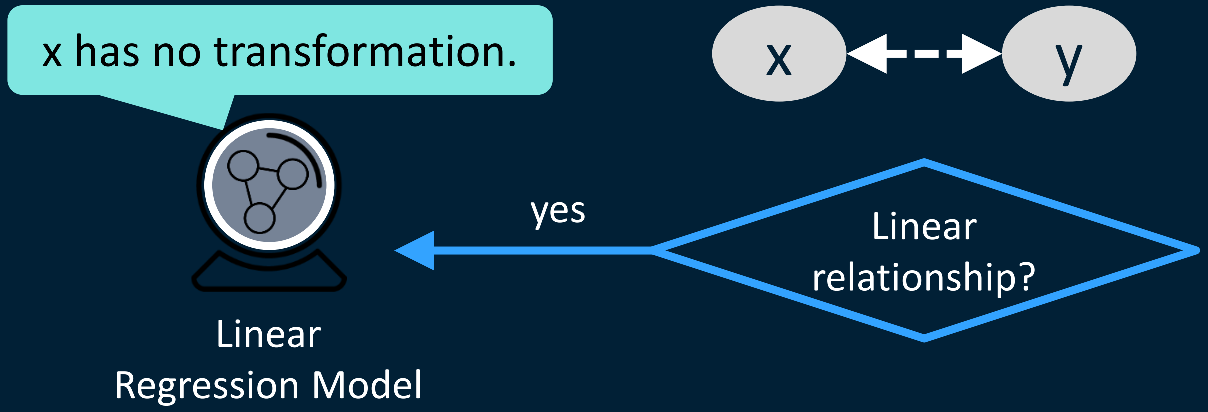
- Finds a global, unique optimum
- Kernel “trick” works well
- General-purpose algorithm that works well with high-dimensional data
- Once the model is trained, prediction is very fast

Disadvantages:

- Training can be very computationally expensive
- Results are strongly dependent on “softening parameter” C
- Results are not interpretable

Neural Networks

Traditional Regression Models



x has no transformation.

x

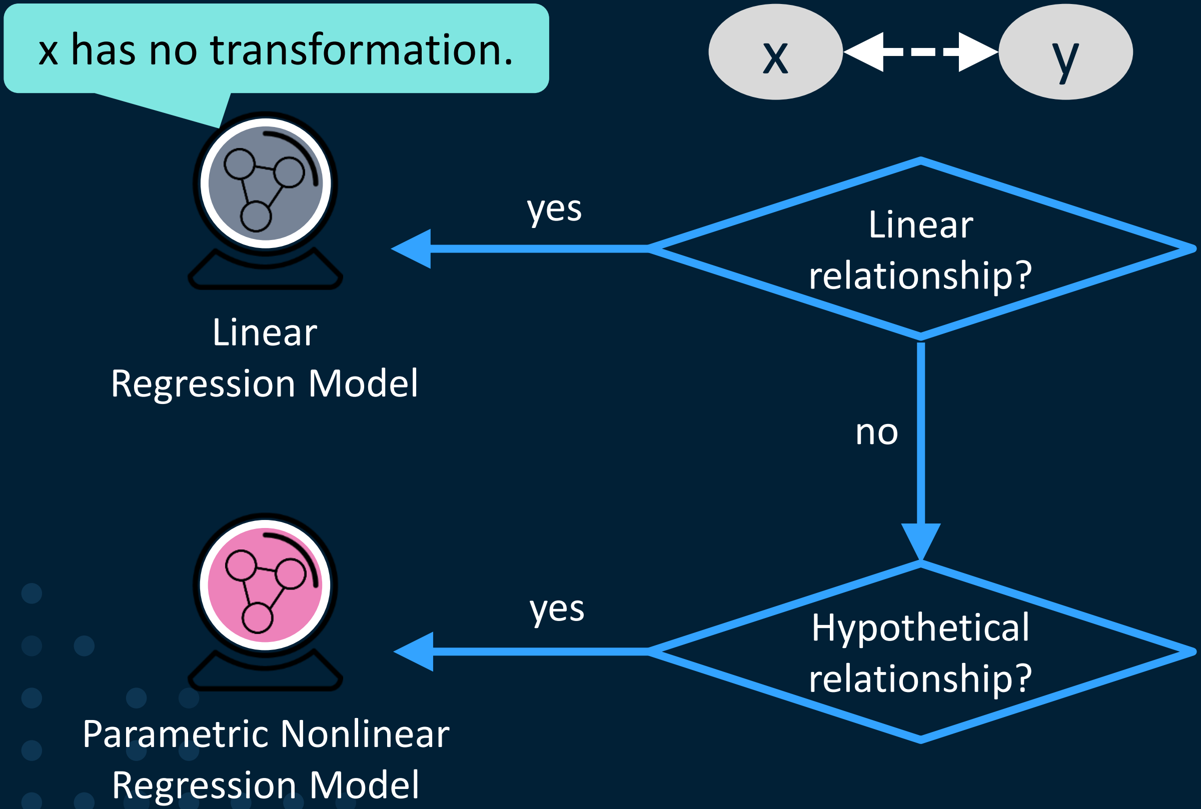
y

Linear relationship?

yes

Linear Regression Model

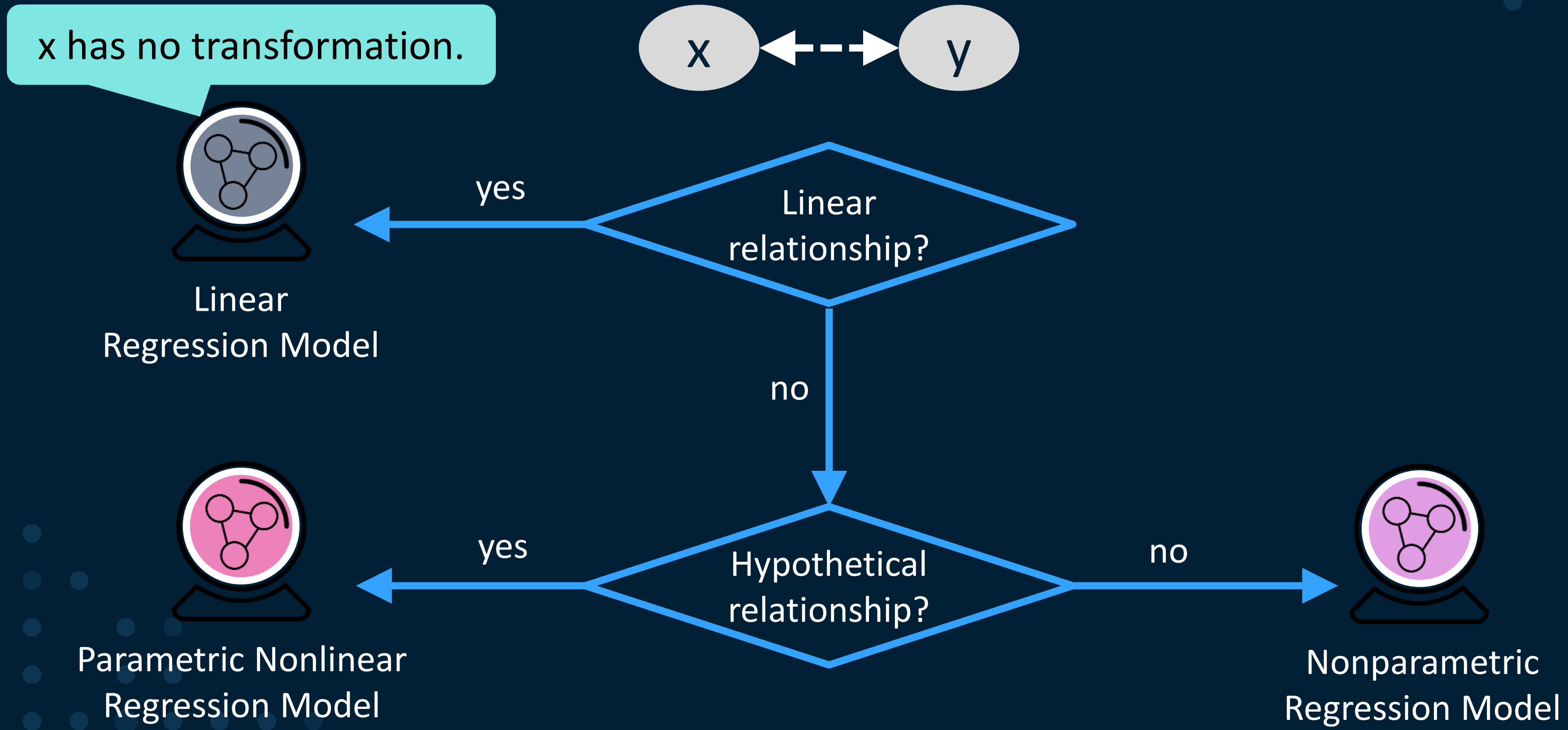
Traditional Regression Models



x has no transformation.

x
y
Linear relationship?
yes
Linear Regression Model
no
Hypothetical relationship?
yes
Parametric Nonlinear Regression Model

Traditional Regression Models



x has no transformation.

x
y
Linear relationship?
yes
Linear Regression Model
no
Hypothetical relationship?
yes
Parametric Nonlinear Regression Model
no
Nonparametric Regression Model

Traditional Regression Models

Parametric Nonlinear Regression Model

Examples

linear in parameters

$$\hat{y} = w_0 + w_1 x + w_2 x^2$$

$$Y = \beta_1 X^{\beta_2} + \varepsilon$$

nonlinear in parameters

Nonparametric Regression Model

no functional form or parameters

Limitations of Traditional Regression Models

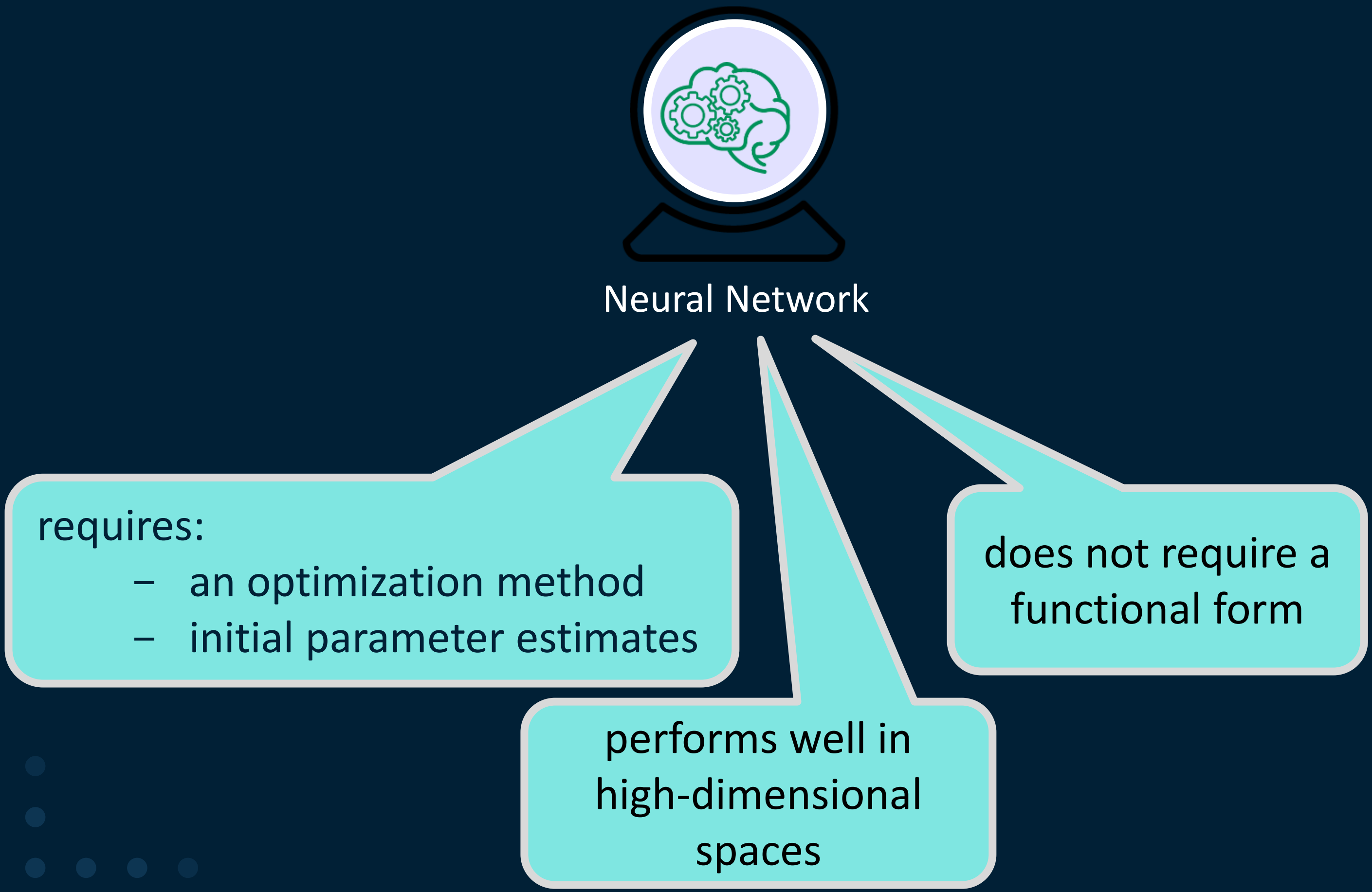
Parametric Nonlinear Regression Model

- more difficult to estimate
- requires:
  - functional form
  - an optimization method
  - good parameter estimates
- curse of dimensionality

Nonparametric Regression Model

- curse of dimensionality

Beyond Traditional Regression: Neural Networks



Neural Network

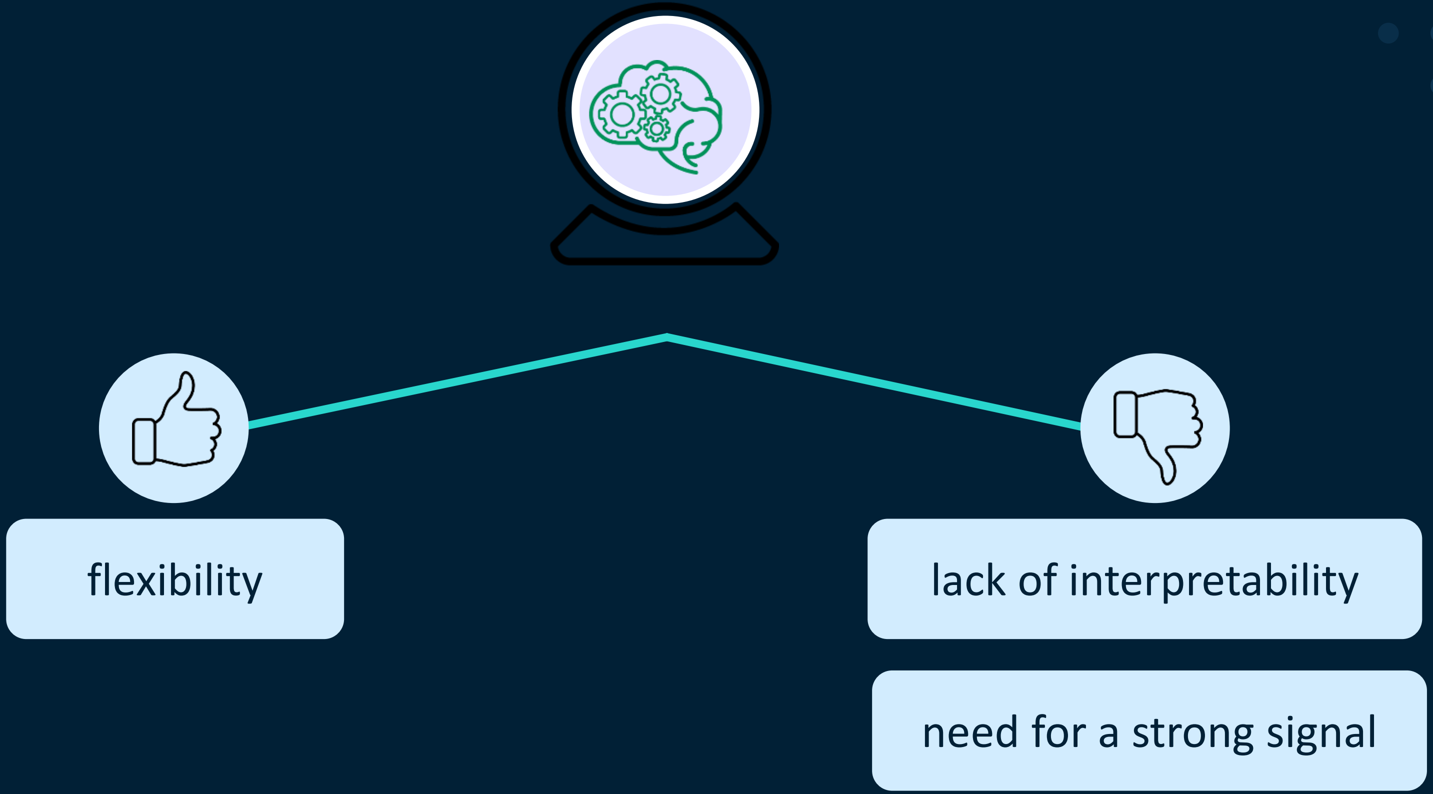
requires:

- an optimization method
- initial parameter estimates

performs well in high-dimensional spaces

does not require a functional form

Advantages and Disadvantages of Neural Networks



flexibility

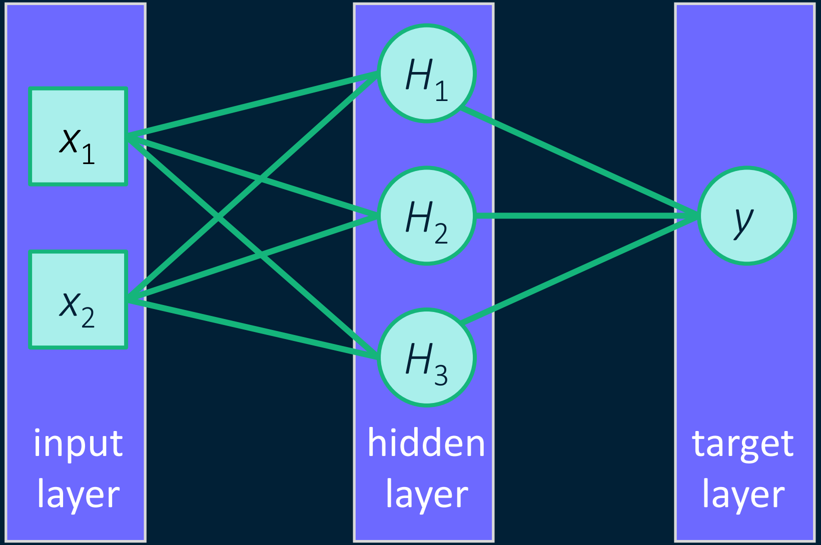
lack of interpretability

need for a strong signal

Response to the Lack of Interpretability Objection

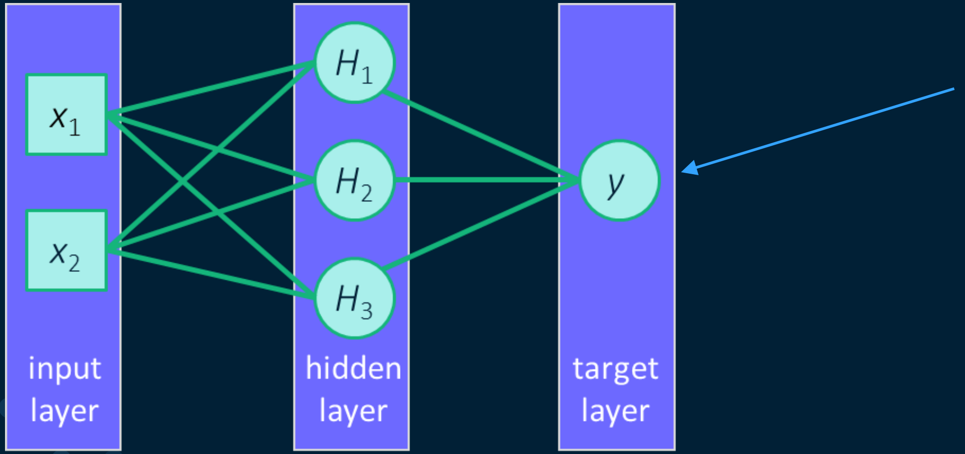
- This is the famous ***black-box objection***, often raised merely to disparage neural networks.
- There are two ways to respond to this objection:
  - by admitting that neural networks are most relevant to pure prediction tasks
  - by applying other modeling techniques, such as decision trees, to try to help “open” the black box

Multilayer Perceptron (MLP)



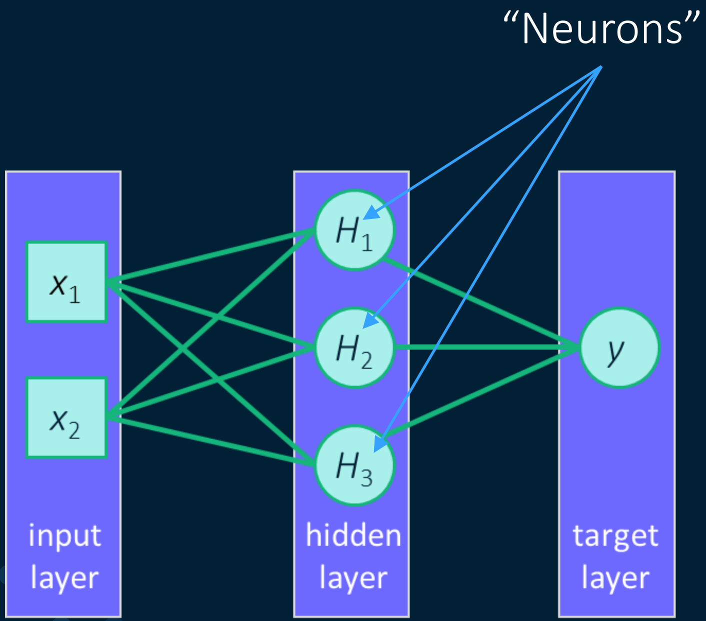
Multilayer Perceptron (MLP)

$$\hat{y} = \widehat{w}_{00} + \widehat{w}_{01} H_1 + \widehat{w}_{02} H_2 + \widehat{w}_{03} H_3$$



A regression model on a set of derived inputs (outputs from the hidden layer)

Multilayer Perceptron (MLP)

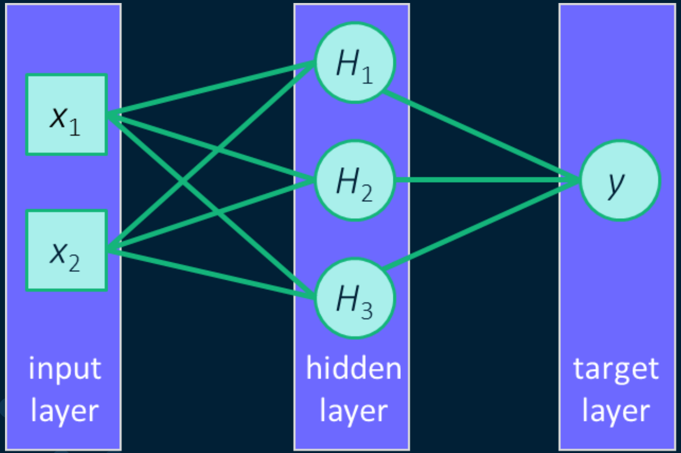


Multilayer Perceptron (MLP)

$$\hat{y} = \hat{w}_{00} + \hat{w}_{01}H_1 + \hat{w}_{02}H_2 + \hat{w}_{03}H_3$$

Bias estimate

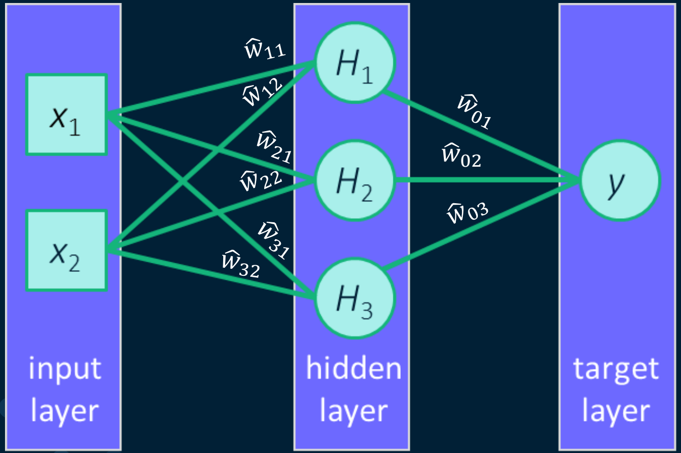
Weight estimates



A regression model on a set of derived inputs (outputs from the hidden layer)

Multilayer Perceptron (MLP)

$$\hat{y} = \hat{w}_{00} + \hat{w}_{01}H_1 + \hat{w}_{02}H_2 + \hat{w}_{03}H_3$$



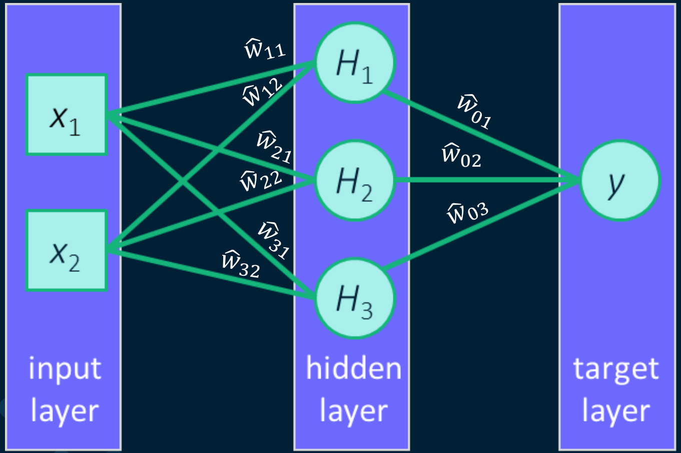
$$H_1 = \tanh(\hat{w}_{10} + \hat{w}_{11}x_1 + \hat{w}_{12}x_2)$$

$$H_2 = \tanh(\hat{w}_{20} + \hat{w}_{21}x_1 + \hat{w}_{22}x_2)$$

$$H_3 = \tanh(\hat{w}_{30} + \hat{w}_{31}x_1 + \hat{w}_{32}x_2)$$

Multilayer Perceptron (MLP)

$$\hat{y} = \hat{w}_{00} + \hat{w}_{01}H_1 + \hat{w}_{02}H_2 + \hat{w}_{03}H_3$$



$$H_1 = \tanh(\hat{w}_{10} + \hat{w}_{11}x_1 + \hat{w}_{12}x_2)$$

$$H_2 = \tanh(\hat{w}_{20} + \hat{w}_{21}x_1 + \hat{w}_{22}x_2)$$

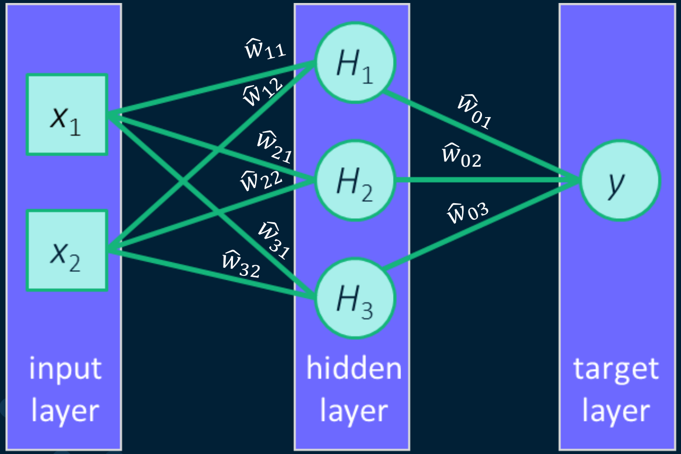
$$H_3 = \tanh(\hat{w}_{30} + \hat{w}_{31}x_1 + \hat{w}_{32}x_2)$$

Activation function

Multilayer Perceptron (MLP)

Can approximate virtually any association

$$\hat{y} = \hat{w}_{00} + \hat{w}_{01}H_1 + \hat{w}_{02}H_2 + \hat{w}_{03}H_3$$



$$H_1 = \tanh(\hat{w}_{10} + \hat{w}_{11}x_1 + \hat{w}_{12}x_2)$$

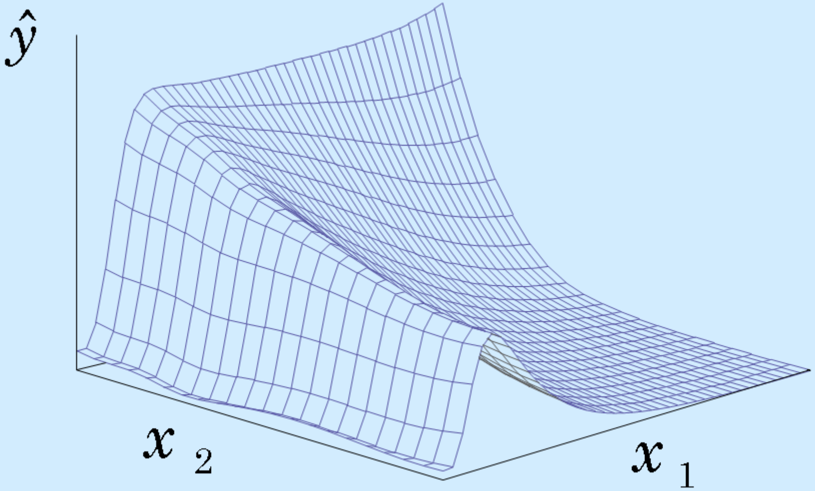
$$H_2 = \tanh(\hat{w}_{20} + \hat{w}_{21}x_1 + \hat{w}_{22}x_2)$$

$$H_3 = \tanh(\hat{w}_{30} + \hat{w}_{31}x_1 + \hat{w}_{32}x_2)$$

Activation function

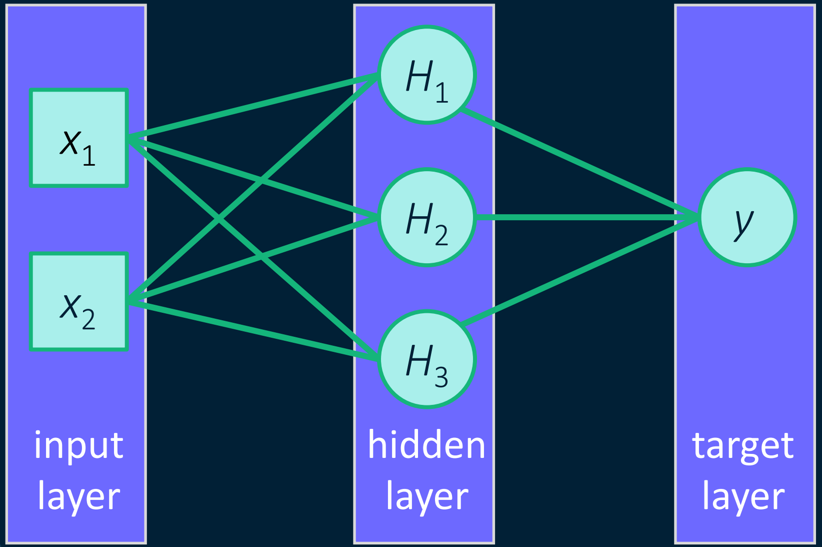
Universal Approximator

Given enough neurons and time, a neural network can model any input/output relationship, to any degree of precision.



Multilayer Perceptron (MLP)

Network Architecture Hyperparameters

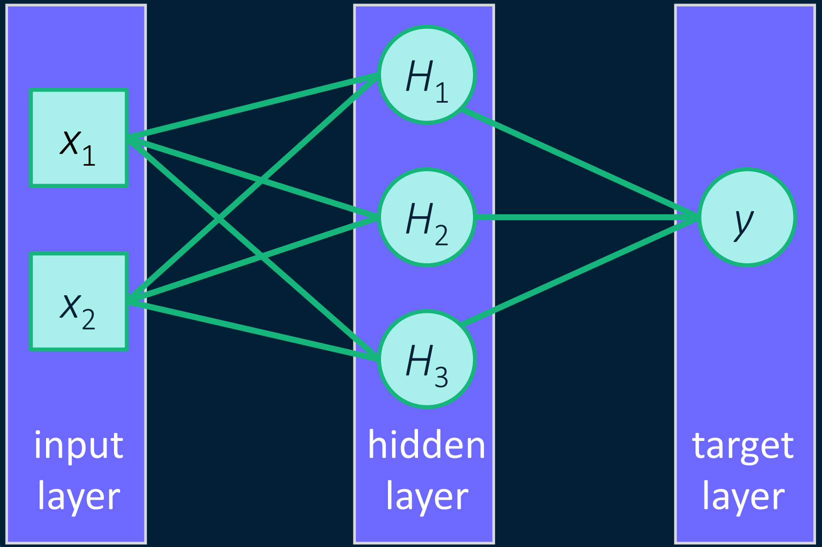


Options:

- Number of hidden layers
- Number of neurons per layer
- Connection types
- Activation functions

Multilayer Perceptron (MLP)

Network Architecture Hyperparameters

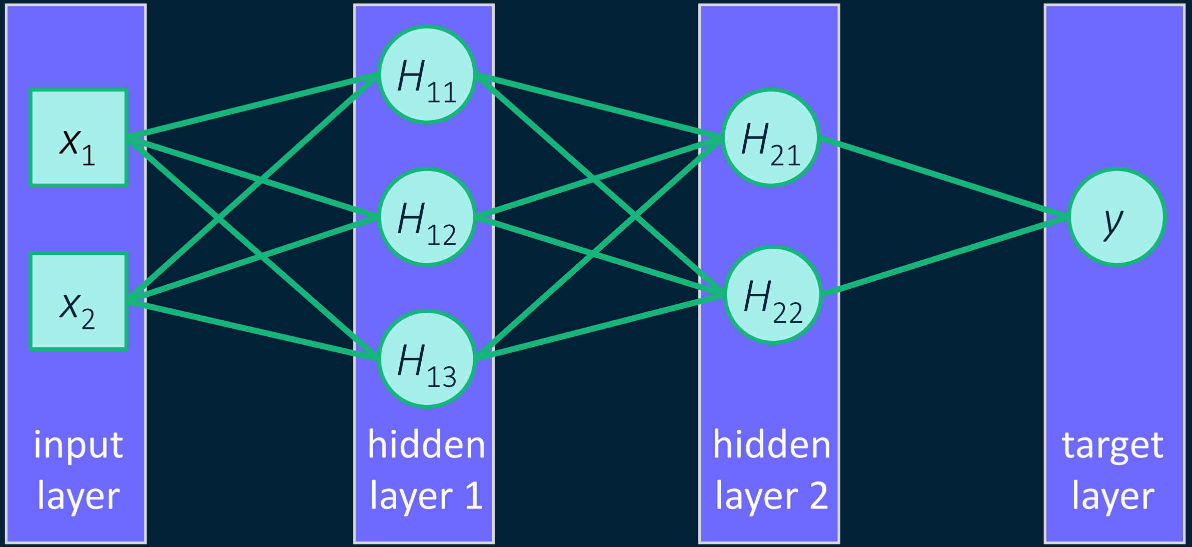


Options:

- Number of hidden layers
- Number of neurons per layer
- Connection types
- Activation functions

Multilayer Perceptron (MLP)

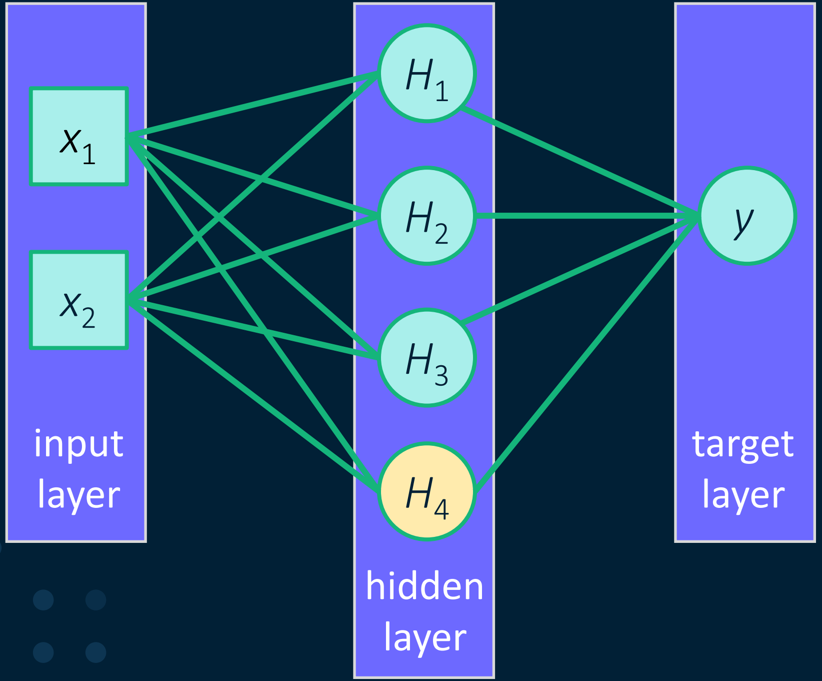
Network Architecture Hyperparameters



- Enables modeling of discontinuous input-output mappings
- Increases the number of weights that need to be estimated

Multilayer Perceptron (MLP)

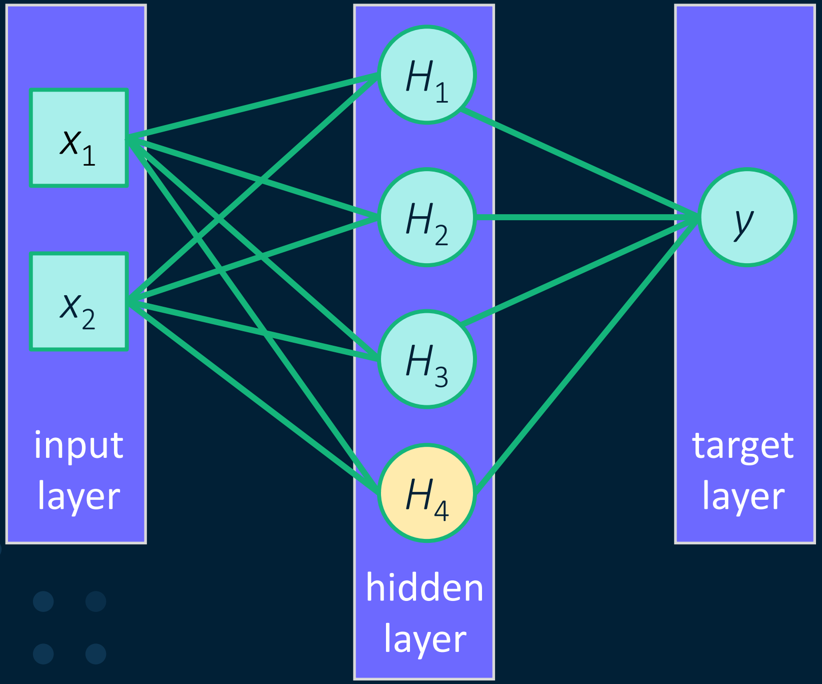
Network Architecture Hyperparameters



- Adding neurons in hidden layers can improve model performance
- Determining optimal number of hidden neurons is more difficult than determining optimal number of hidden layers

Multilayer Perceptron (MLP)

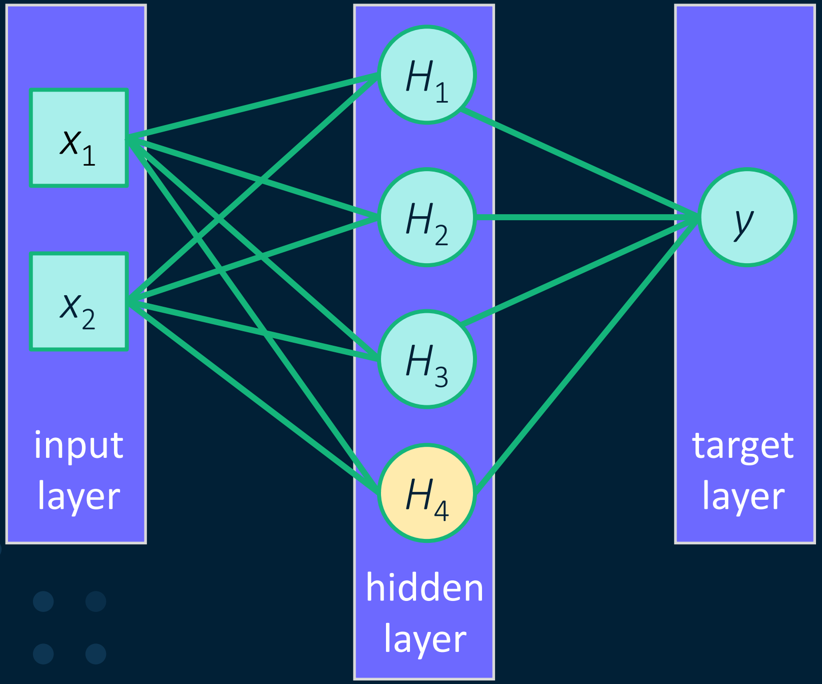
Network Architecture Hyperparameters



Hidden Neurons	Result
Too many	- Models noise - Fails to generalize
Too few	- Fails to capture the signal - Fails to generalize

Multilayer Perceptron (MLP)

Network Architecture Hyperparameters



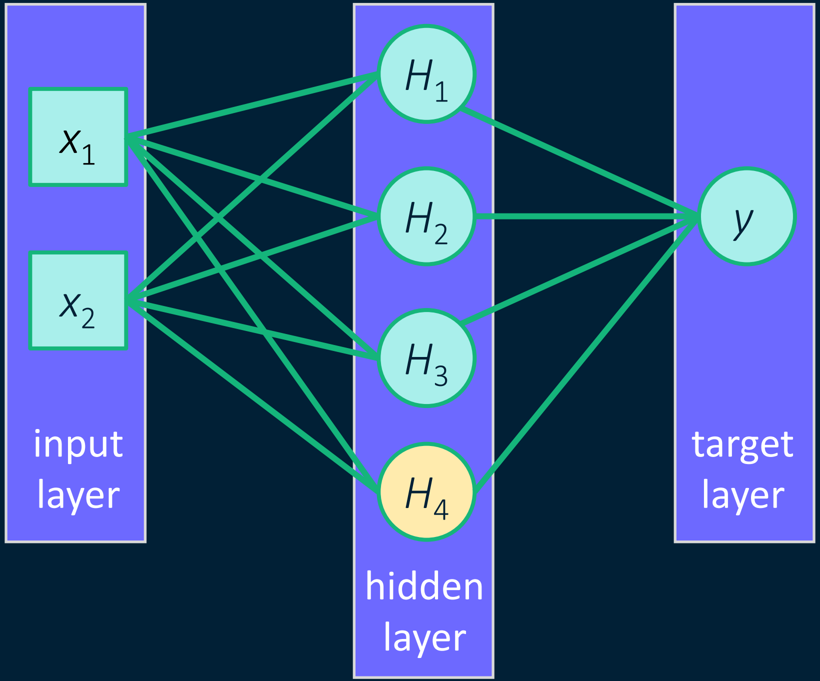
Guidelines:

- Number of hidden neurons in first layer should be approximate twice the number of input dimensions

Multilayer Perceptron (MLP)

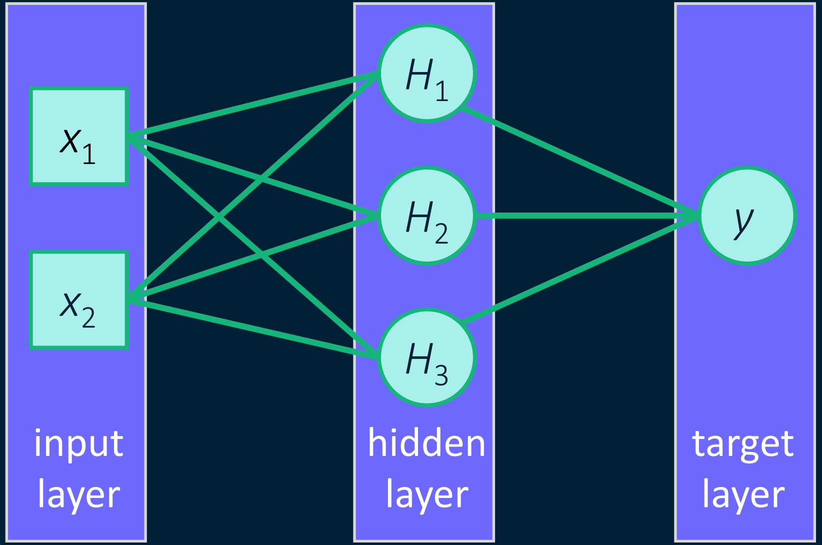
Network Architecture Hyperparameters

- Unlike tree-based models, neural networks cannot “select” inputs
- Therefore, it is a good practice to also focus on variable selection prior to finalizing the components of the input layer



Multilayer Perceptron (MLP)

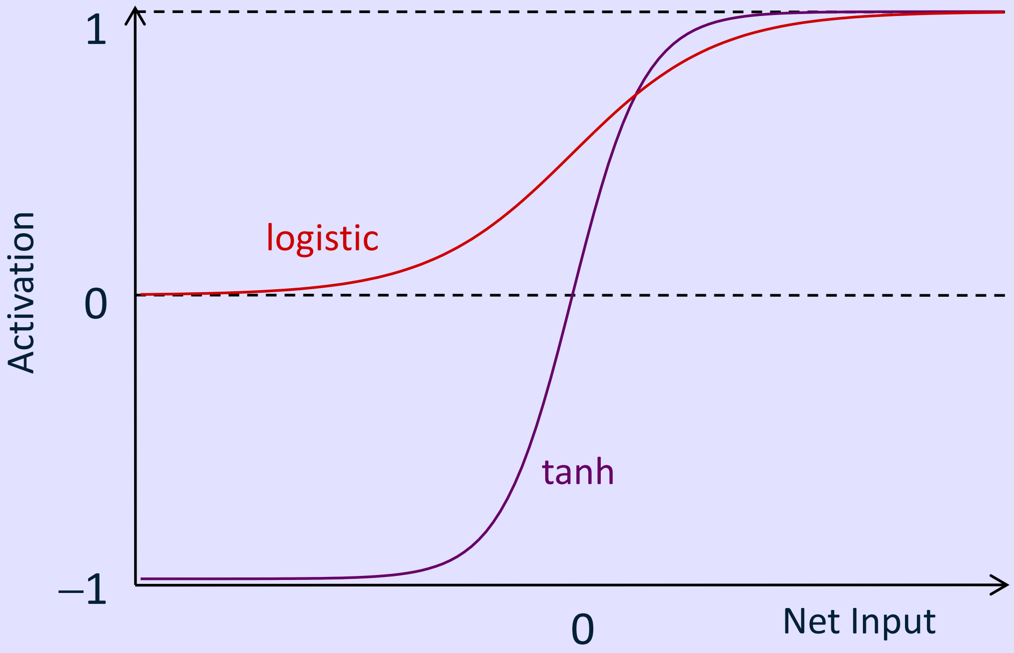
Network Architecture Hyperparameters



Hidden layer is the key to modeling non-linearities

- "Activation function" introduces the nonlinearity
- Original neural networks used the logit function as the activation functions
- Several other options are used today.
- SAS uses the hyperbolic tangent as the default activation function

Activation Functions



Activation Function Examples

Function	Plot	Equation	Range
Exponential		$f(x) = e^x$	$[0, \infty)$
Identity		$f(x) = x$	$(-\infty, \infty)$
Logistic		$f(x) = \dfrac{1}{1 + e^{-x}}$	$(0,1)$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$	$[0, \infty)$
Sine		$f(x) = \sin(x)$	$[-1,1]$
Softplus		$f(x) = \ln(1 + e^x)$	$[0, \infty)$
Hyperbolic Tangent (Tanh)		$f(x) = \dfrac{(e^x - e^{-x})}{(e^x + e^{-x})}$	$(-1,1)$

Optimize Model Complexity

Weight Regularization

$$\hat{y} = \widehat{w}_{00} + \widehat{w}_{01}H_1 + \widehat{w}_{02}H_2 + \widehat{w}_{03}H_3$$

$$H_1 = \tanh(\widehat{w}_{10} + \widehat{w}_{11}x_1 + \widehat{w}_{12}x_2)$$

$$H_2 = \tanh(\widehat{w}_{20} + \widehat{w}_{21}x_1 + \widehat{w}_{22}x_2)$$

$$H_3 = \tanh(\widehat{w}_{30} + \widehat{w}_{31}x_1 + \widehat{w}_{32}x_2)$$

Optimize Model Complexity

$$\hat{y} = \widehat{w}_{00} + \widehat{w}_{01}H_1 + \widehat{w}_{02}H_2 + \widehat{w}_{03}H_3$$

$$H_1 = \tanh(\widehat{w}_{10} + \widehat{w}_{11}x_1 + \widehat{w}_{12}x_2)$$

$$H_2 = \tanh(\widehat{w}_{20} + \widehat{w}_{21}x_1 + \widehat{w}_{22}x_2)$$

$$H_3 = \tanh(\widehat{w}_{30} + \widehat{w}_{31}x_1 + \widehat{w}_{32}x_2)$$

Optimizing the complexity of a neural network involves controlling the magnitude of the weights

- If the weights grow too large, the model will be overfit
- Large weights cause the activation functions to become too steep and “adapt” to noise
- The two methods of controlling overfitting are called “weight decay” and “early stopping”

Optimize Model Complexity

Weight Decay

- Recall that when estimating weights we attempt to minimize the error function:

$$\mathit{Error\ Function} = -2 * \left[\sum \log(\hat{p}_i) + \sum \log(1-\hat{p}_i)\right]$$

- Regularization adds "penalty terms" to this minimization:
  - L1 Regularization: Minimize $\frac{1}{n}\sum_{i=1}^{n}(\,EF_i + \lambda|w_i|)$
  - L2 Regularization: Minimize $\frac{1}{n}\sum_{i=1}^{n}(\,EF_i + \lambda|w_i|^2)$

Hyperparameter Tuning

Hyperparameters include:

- Number of hidden layers
- Number of neurons per layer
- Activation function
- Network learning hyperparameters

“Wrap Up” Topics

Topics

- Correlated predictors
- Skew in predictor variables
- “Dummy Variable Trap”
- Degrees of freedom
- Multinomial logistic regression

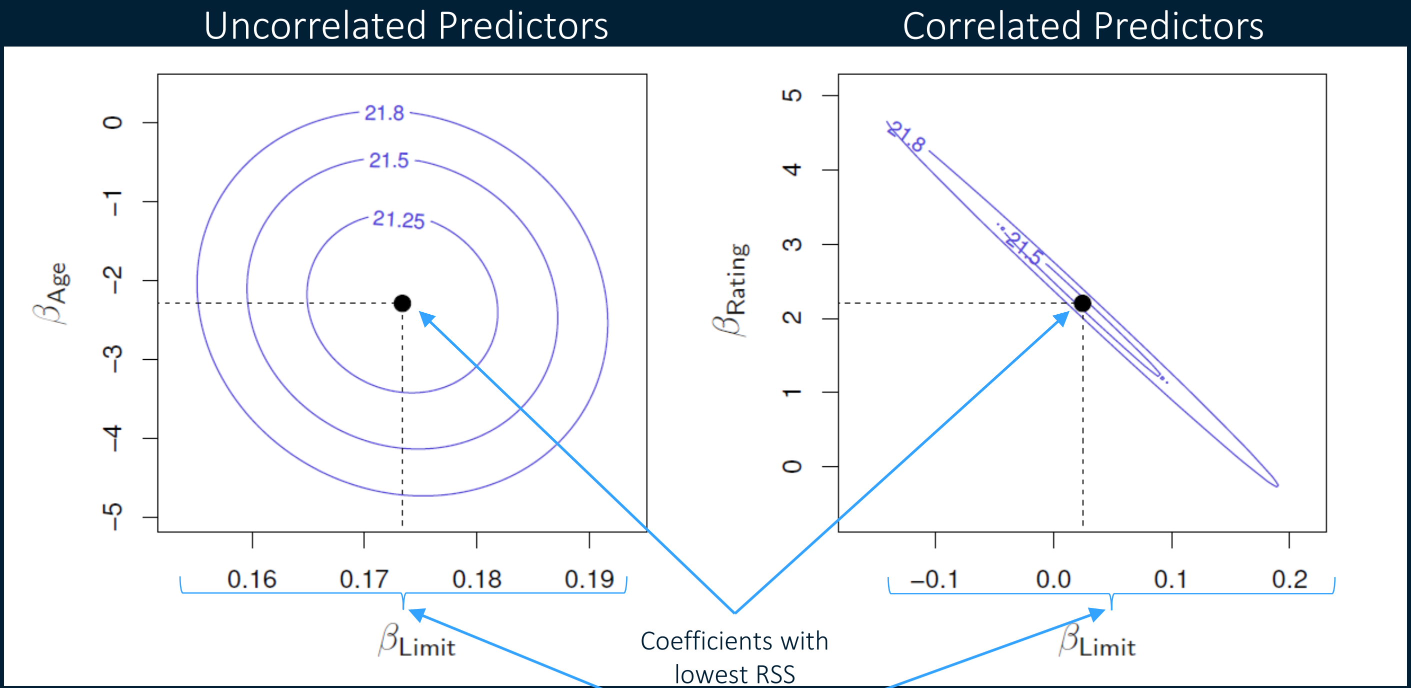
Correlated Predictors and Linear Models

Issues

- Inference is significantly “clouded”
  - Confounding (one predictor is correlated with both the predictor and the response)
- Increases the standard error of the model coefficient:
  - Increases model variance
  - Decreases the power of the hypothesis test (probability of correctly detecting a non-zero coefficient)

Correlated Predictors and Linear Models

RSS Contour Plots



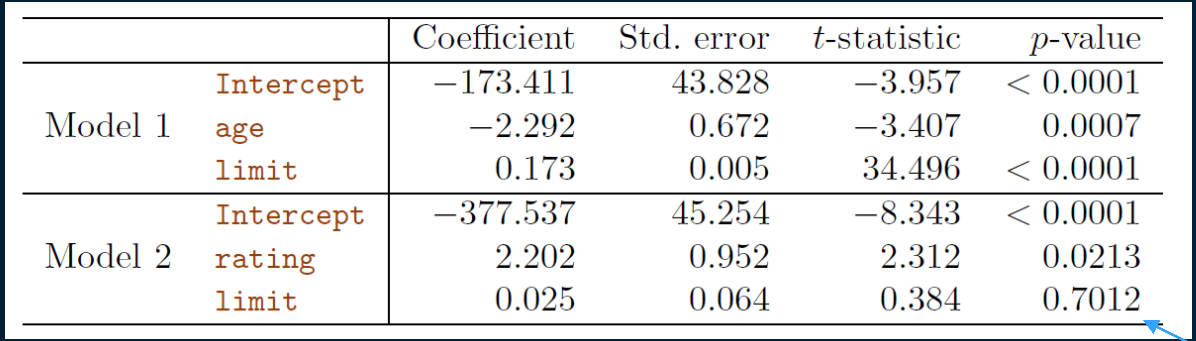
Coefficients with lowest RSS

ISLR Figure 3.14

Axes scaled to include possible coefficients up to 4 standard errors

Correlated Predictors and Linear Models

- Collinearity causes the standard error of the coefficient estimates



		Coefficient	Std. error	t-statistic	p-value
Model 1	`Intercept`	−173.411	43.828	−3.957	< 0.0001
	`age`	−2.292	0.672	−3.407	0.0007
	`limit`	0.173	0.005	34.496	< 0.0001
Model 2	`Intercept`	−377.537	45.254	−8.343	< 0.0001
	`rating`	2.202	0.952	2.312	0.0213
	`limit`	0.025	0.064	0.384	0.7012

ISLR Table 3.11

The importance of the limit predictor is masked by the collinearity

Correlated Predictors and Linear Models

Approaches to resolving mutlicorrelation issues:

- Drop one of the problematic variables
- Combine two or more correlated variables into a single predictor
  - PCA is one technique

Both approaches are risky without using some knowledge of the data meanings and sources

Skewed Variables in Linear Regression

- There is no assumption of normality of predictors or response in a linear model
  - The only assumption is that the residuals are normal
- However, highly skewed variables can cause other problems
  - They can influence the distribution of the residuals, making them non-normal
  - They can significantly complicate the identification of high-leverage observations

When to Consider a Non-Linear Transformation?

- When underlying scientific theory indicates that the expected relationship is nonlinear
  - For example, if you have reason to believe that the true relationship is multiplicative, taking log transforms allows the model to be treated as linear
- Residuals have a skewed distribution
- Heteroscedasticity
- To simplify a model
  - For example, sometimes it can reduce or simply interaction effects
- Log transforms can yield more interpretable coefficients

Linear Regression Coefficient Interpretations

With everything else held equal (using log_{10}):

- Y as a function of X: a one unit increase in X leads to a β increase/decrease in Y
- Log Y as a function of log X: a 1% increase in X leads to a $\beta\%$ increase/decrease in Y
- Log Y as a function of X: a one unit increase in X leads to a $\beta * 100\%$ increase/decrease in Y
- Y as a function of Log X: a 1% increase in X leads to a $\beta/100$ increase/decrease in Y

Reasons NOT to Use a Transform

- Make outliers not look like outliers
- Because all the data are positive (positivity often implies skewness but it doesn’t have to)
- Because the software automatically does it

Degrees of Freedom

- Many statistical tests and probability distributions include a value referred to as "degrees of freedom"
- Concept applied to statistics calculated from sample data
  - Refers to the number of values free to vary after the sample statistic has been established
  - For example, if you are calculating a sample mean of 10 values, the Degrees of Freedom is 9 (once you know 9 samples and the sample mean, the 10th can be calculated and is not free to vary)

Degrees of Freedom

- Degrees of freedom are used in many statistical tests to avoid bias in the sample results, for example, sample standard variance:

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

Degrees of Freedom

“Contingency Table”

- A contingency table contains counts of observations in two categories:

	Category A	Category B	Category C	Total
Category I				xx
Category II				xx
Total	xx	xx	xx	

- What would be the formula for the number of degrees of freedom?

Degrees of Freedom

Generally, not a concept data scientists are often concerned with

- Statistical tests are used sparingly in data science
- Data sizes are large enough it is generally insignificant ($\frac{1}{n-1} \approx \frac{1}{n}$ for large n)

Degrees of Freedom

Intuition

- Sample variance is an estimate of the true population variance:

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^2$$

Why the difference?

On average, which is any given X_i going to be closer to, $\bar{X}$ or μ?

Degrees of Freedom

- One place it comes into play is when converting a factor variable into indicator or “dummy” variables
- Consider the “origin” field in the cars dataset which can have three possible values (“Asia”, “Europe”, and “USA”)
- There are only 2 (not 3) degrees of freedom and there will be perfect multicollinearity (and infinite VIF) for each dummy variable
  - There will not be a unique solution

Multinomial Logistic Regression

Overview

- Extension of logistic regression for multiple categories
- Select a single class to serve as the baseline (here, we select K):

$$P(Y = k|X = x) = \frac{e^{(\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p)}}{1 + e^{(\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p)}}$$

for k = 1, …, K-1, and

$$P(Y = K|X = x) = \frac{1}{1 + e^{(\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p)}}$$

Multinomial Logistic Regression

Overview

Thus, for $k = 1 \dots K - 1$:

$$\log\left(\frac{P(Y = k|X = x)}{P(Y = K|X = x)}\right) = \beta_{k0} + \beta_{k1}x_1 + \cdots + \beta_{kp}x_p$$

Note: the selection of the K_{th} class is arbitrary and unimportant

Multinomial Logistic Regression

Inference

However, your K_{th} class becomes the baseline and then a one-unit increase in X_j is associated with a β_{kj} increase in the log odds of event k over event K

Final Exam Logistics

Final Exam

Logistics

- Final exam will be administered in class on Thursday, August 11
  - 4:00PM – 6:00PM
  - WILL BE ENTIRELY REMOTE
- Exam will be open-book/open-notes but must be done individually

Final Exam

Format

- Final exam will consist of short-answer questions to test your understanding of the theoretical concepts presented in module 5-9
  - Interpretation of model outputs and visualizations
  - Simple calculations that can be done manually of using Excel
- Exam will be a PowerPoint file
  - Similar to homework assignments, you will download the PowerPoint exam file, enter your responses, save it to a PDF file format and upload it to Gradescope

Sample Theory Questions

Module 5 – Linear Model Selection and Regularization

Sample Theory Questions

- Manual calculation of Adjusted R2, Cp, AIC, BIC
- Difference between Lasso and Ridge Regression
- Why use the third partition
- General data preparation best practices
- * No questions on dimension reduction

Module 6 – Linear Classification Models

Sample Theory Questions

- Interpreting logit function
- Maximum Likelihood Estimation (MLE) - basic operations
- Interpreting and calculating odds ratios
- Interpreting logistic regression coefficients
- Confounding - basic concept
- Classification model assessment statistics - calculation and interpretation
    - TPR/FPR/TNR/FNR/Sensitivity/Specificity/Confusion matrix

Module 6 – Linear Classification Models

Sample Theory Questions

- ROC curve
  - Construction and interpretation
- Lift curve and lift as a concept
  - Construction and interpretation
- Event-based sampling - what it is and why/how to use it
- Multinomial logistic regression - definition and simple calculations

Module 7 – Generalized Linear Models and Poisson Regression

Sample Theory Questions

- Deciding when to use Poisson regression
- Deciding when to treat a predictor as a category vs a measure
- Interpreting residuals for a Poisson Regression model
- Interpreting Poisson regression coefficients
- GLM – distribution family and link function for three types of GLMs (linear, logistic, Poisson)

Module 8 – Moving Beyond Linearity

Sample Theory Questions

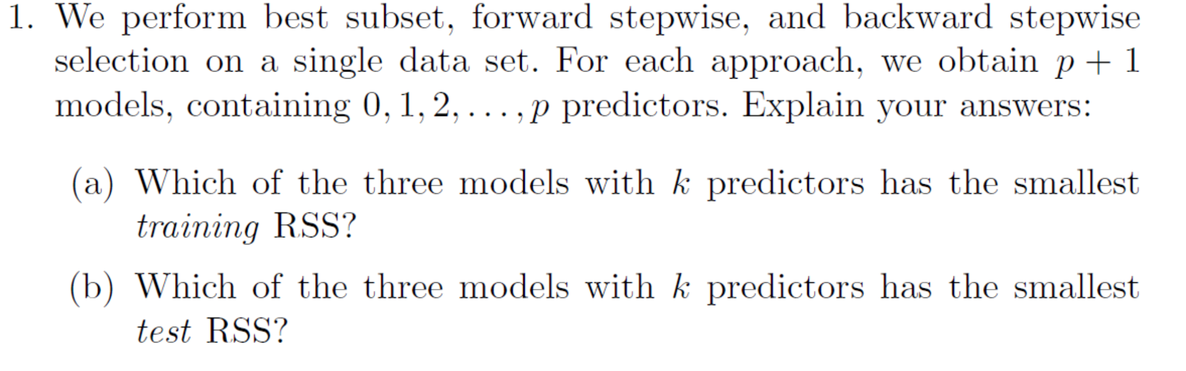
- Basis function definition
- Step functions - definition and implementation
- Piecewise polynomials - definition and implementation
- General additive models - definition

Module 9 – Tree-Based Models

Sample Theory Questions

- Purity calculations
- Tree pruning approaches
- Cost complexity pruning
- Advantages and disadvantages of trees
- Ensemble models - definition and types (averaging and boosting)
- Random forest - basic approach
- AdaBoost algorithm - basic operation
- Gradient Boosting algorithm - basic operation

Sample Questions – Module 5



1. We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain $p + 1$ models, containing $0, 1, 2, \ldots, p$ predictors. Explain your answers:

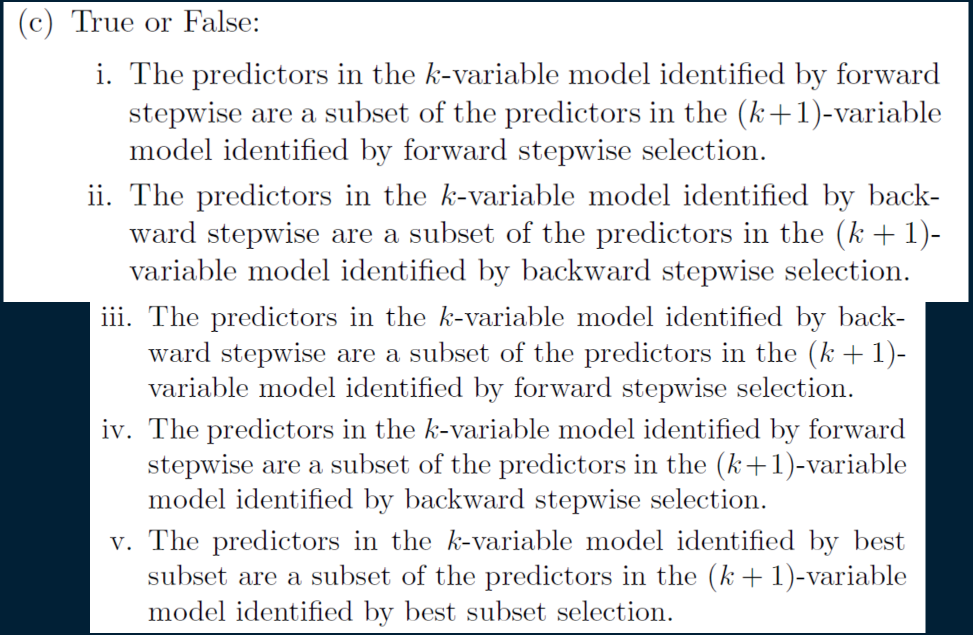
   (a) Which of the three models with k predictors has the smallest *training* RSS?

   (b) Which of the three models with k predictors has the smallest *test* RSS?

Best subset

Can’t say definitievely

Sample Questions – Module 5



(c) True or False:

i. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by forward stepwise selection.

True

ii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by backward stepwise selection.

True

iii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by forward stepwise selection.

False

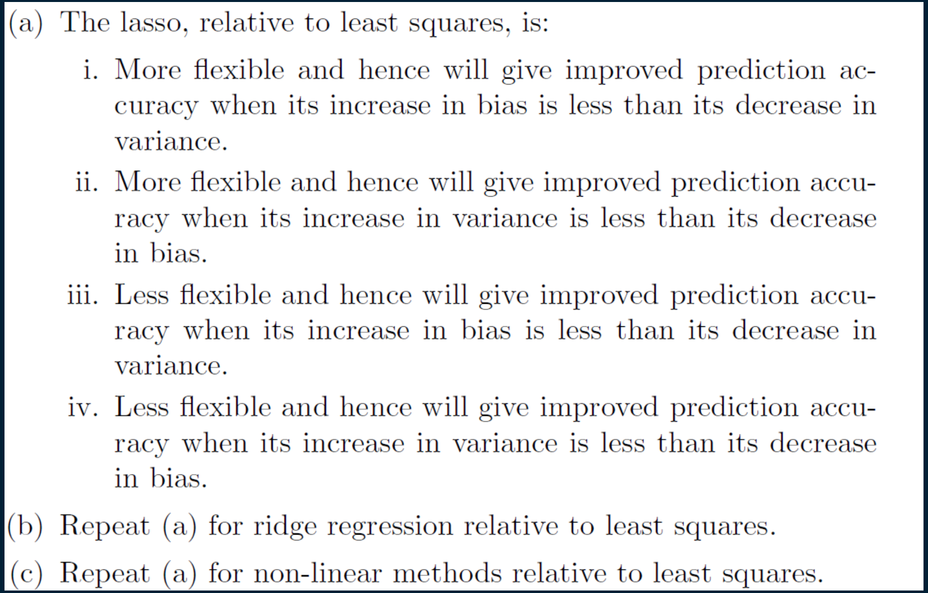
iv. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by backward stepwise selection.

False

v. The predictors in the k-variable model identified by best subset are a subset of the predictors in the $(k+1)$-variable model identified by best subset selection.

False

Sample Questions – Module 5



(a) The lasso, relative to least squares, is:

  i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

  ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

  iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

  iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

(b) Repeat (a) for ridge regression relative to least squares.

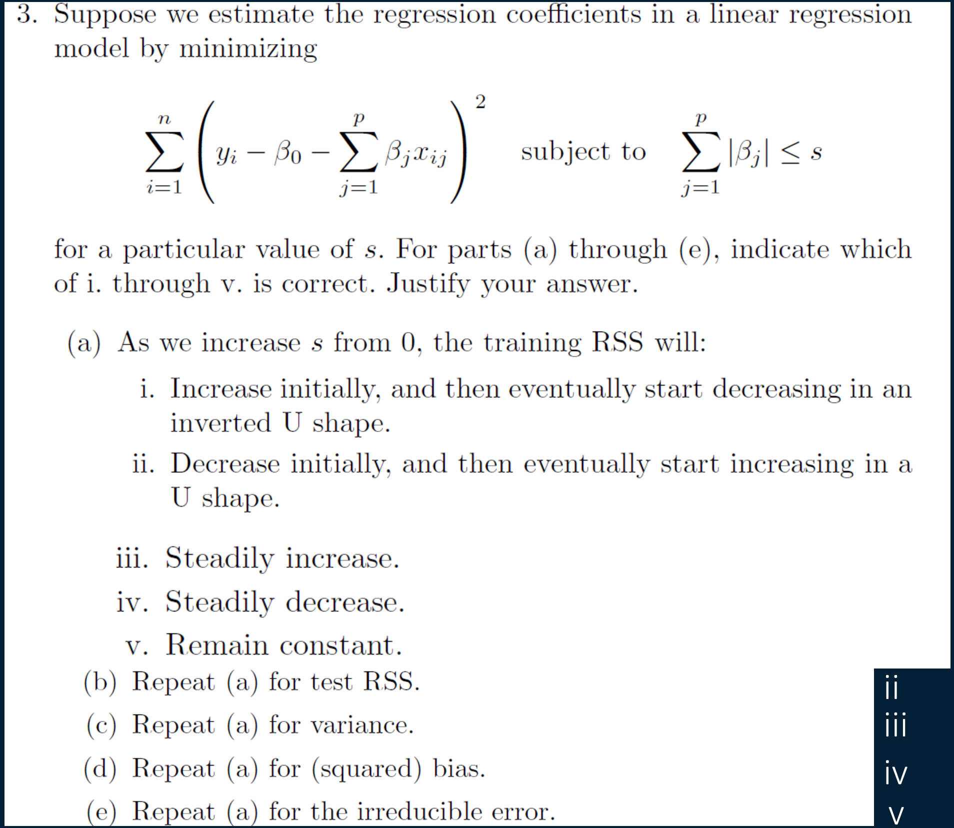
(c) Repeat (a) for non-linear methods relative to least squares.

iii

iii

i

Sample Questions – Module 5



3. Suppose we estimate the regression coefficients in a linear regression model by minimizing

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p}|\beta_j| \leq s$$

for a particular value of s. For parts (a) through (e), indicate which of i. through v. is correct. Justify your answer.

(a) As we increase s from 0, the training RSS will:

i. Increase initially, and then eventually start decreasing in an inverted U shape.

ii. Decrease initially, and then eventually start increasing in a U shape.

iii. Steadily increase.

iv. Steadily decrease.

v. Remain constant.

(b) Repeat (a) for test RSS.

(c) Repeat (a) for variance.

(d) Repeat (a) for (squared) bias.

(e) Repeat (a) for the irreducible error.

iv

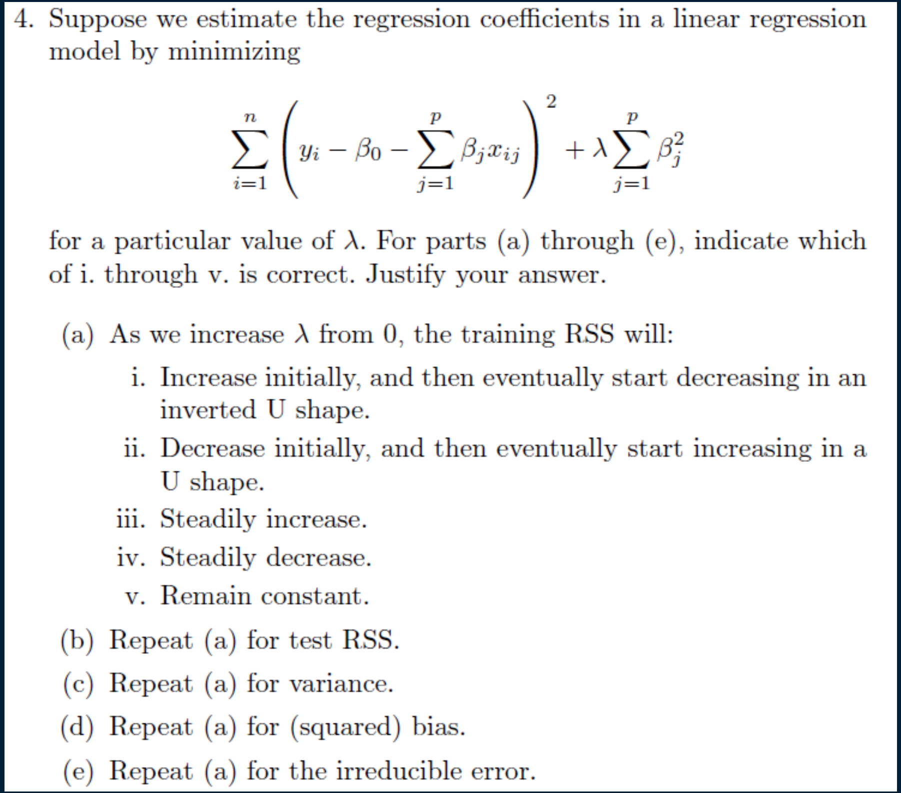
ii

iii

iv

v

Sample Questions – Module 5



4. Suppose we estimate the regression coefficients in a linear regression model by minimizing

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2$$

for a particular value of λ. For parts (a) through (e), indicate which of i. through v. is correct. Justify your answer.

iii

(a) As we increase λ from 0, the training RSS will:

i. Increase initially, and then eventually start decreasing in an inverted U shape.

ii. Decrease initially, and then eventually start increasing in a U shape.

iii. Steadily increase.

iv. Steadily decrease.

v. Remain constant.

(b) Repeat (a) for test RSS.

ii

(c) Repeat (a) for variance.

iv

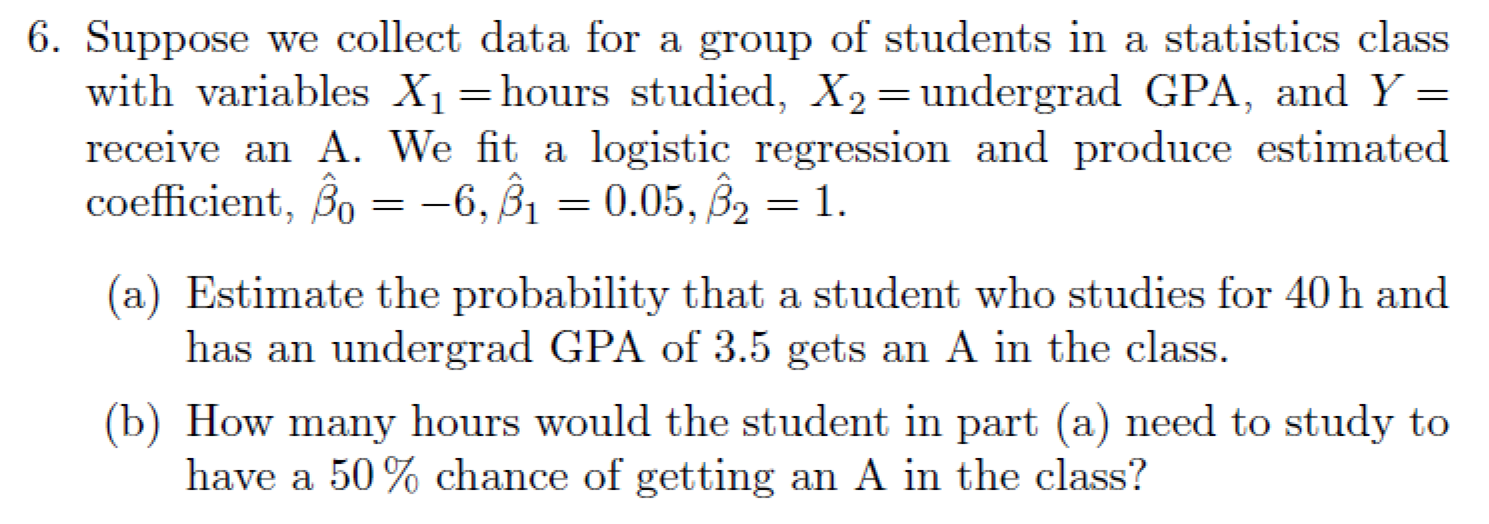
(d) Repeat (a) for (squared) bias.

iii

(e) Repeat (a) for the irreducible error.

v

Sample Questions – Module 6

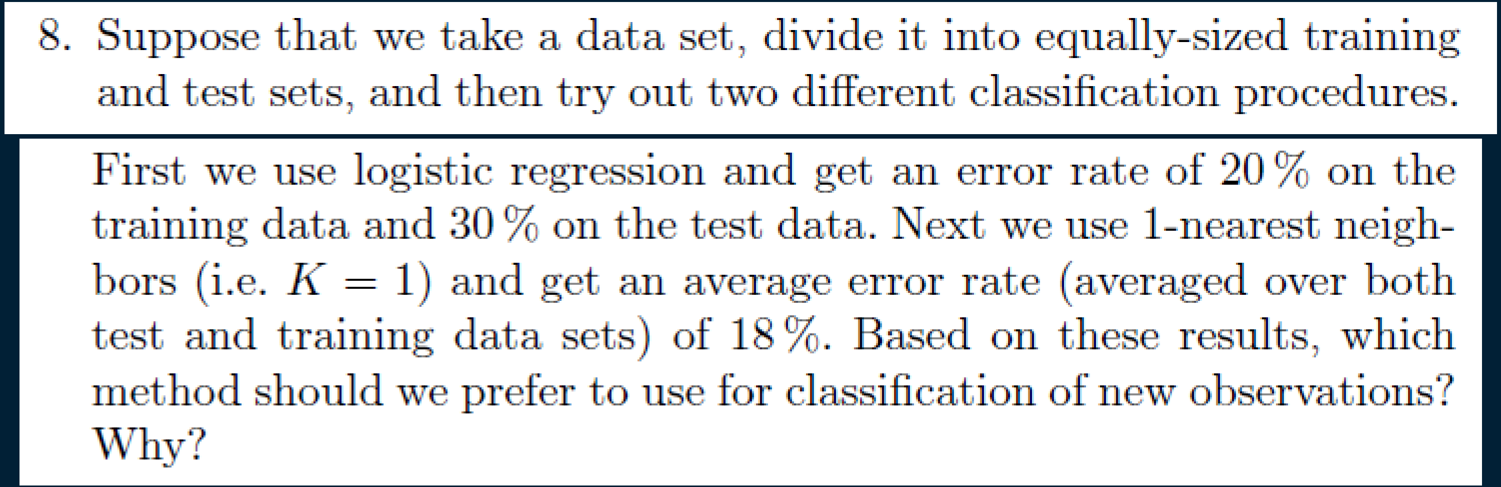


6. Suppose we collect data for a group of students in a statistics class with variables $X_1 =$ hours studied, $X_2 =$ undergrad GPA, and $Y =$ receive an A. We fit a logistic regression and produce estimated coefficient, $\hat{\beta}_0 = -6, \hat{\beta}_1 = 0.05, \hat{\beta}_2 = 1$.

   (a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

   (b) How many hours would the student in part (a) need to study to have a 50 % chance of getting an A in the class?

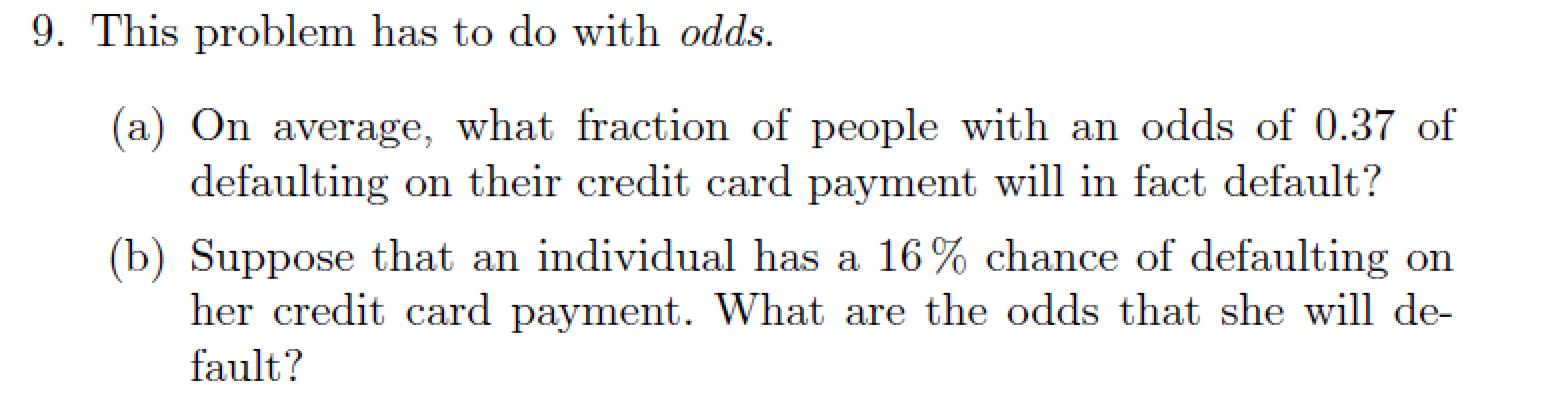
Sample Questions – Module 6



8. Suppose that we take a data set, divide it into equally-sized training and test sets, and then try out two different classification procedures.

   First we use logistic regression and get an error rate of 20 % on the training data and 30 % on the test data. Next we use 1-nearest neighbors (i.e. $K = 1$) and get an average error rate (averaged over both test and training data sets) of 18 %. Based on these results, which method should we prefer to use for classification of new observations? Why?

Sample Questions – Module 6



9. This problem has to do with *odds*.

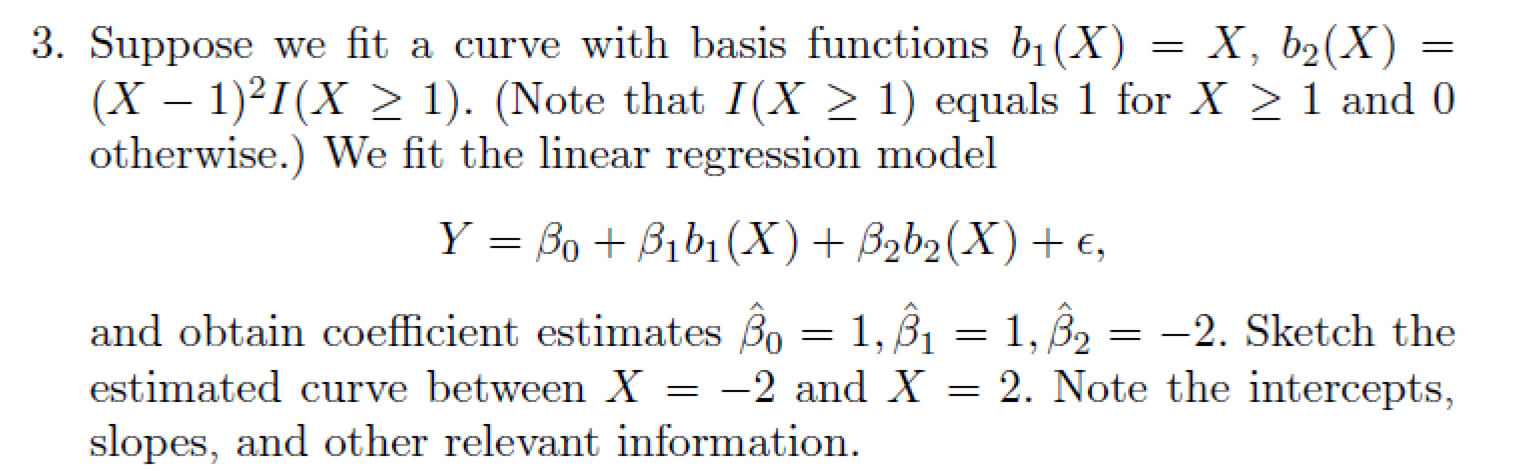
   (a) On average, what fraction of people with an odds of 0.37 of defaulting on their credit card payment will in fact default?

   (b) Suppose that an individual has a 16 % chance of defaulting on her credit card payment. What are the odds that she will default?

$$\frac{0.37}{1.37}$$

$$\frac{0.16}{(1-0.16)}$$

Sample Questions – Module 8

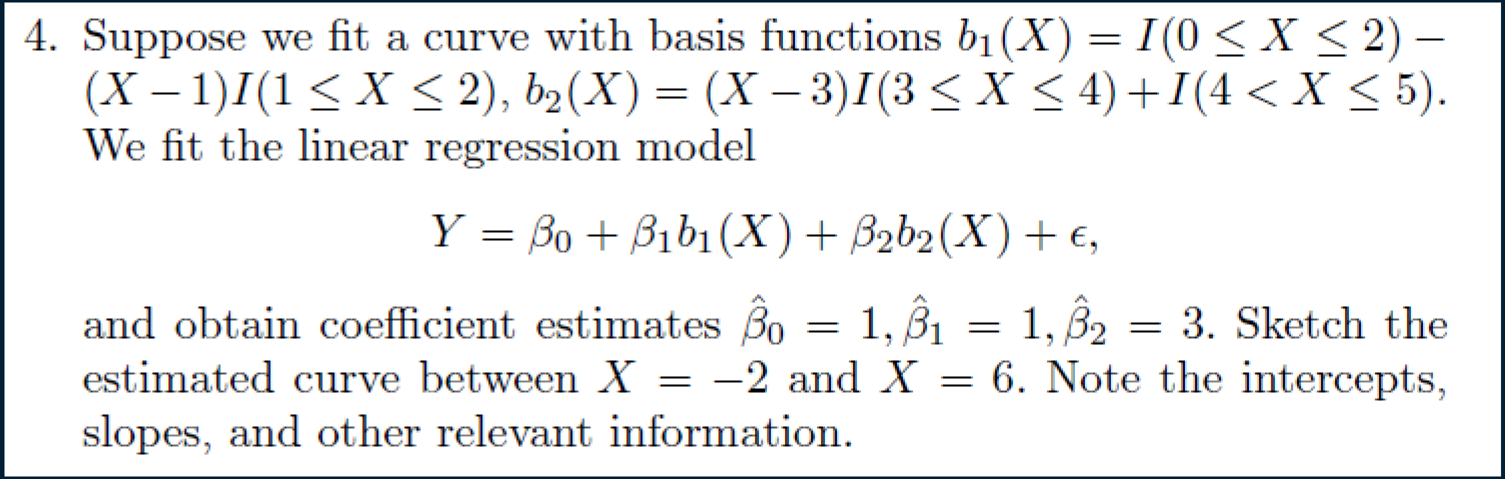


3. Suppose we fit a curve with basis functions $b_1(X) = X$, $b_2(X) = (X-1)^2 I(X \geq 1)$. (Note that $I(X \geq 1)$ equals 1 for $X \geq 1$ and 0 otherwise.) We fit the linear regression model

$$Y = \beta_0 + \beta_1 b_1(X) + \beta_2 b_2(X) + \epsilon,$$

and obtain coefficient estimates $\hat{\beta}_0 = 1, \hat{\beta}_1 = 1, \hat{\beta}_2 = -2$. Sketch the estimated curve between $X = -2$ and $X = 2$. Note the intercepts, slopes, and other relevant information.

Sample Questions – Module 8

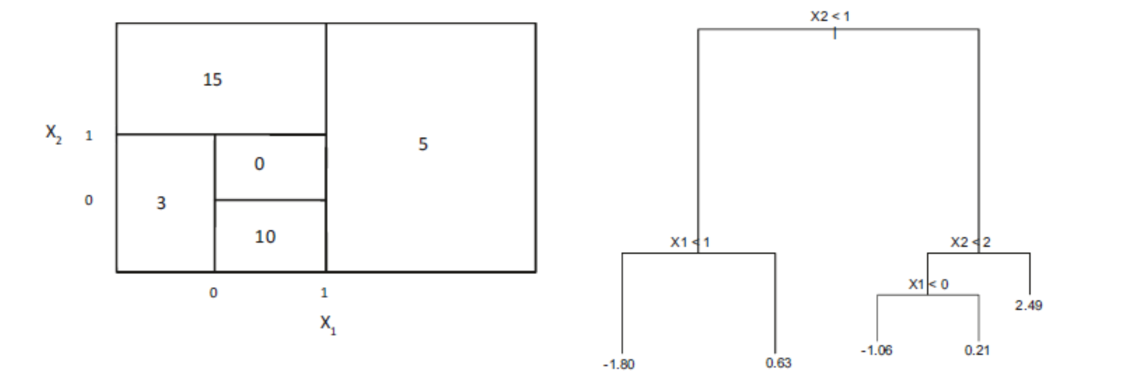


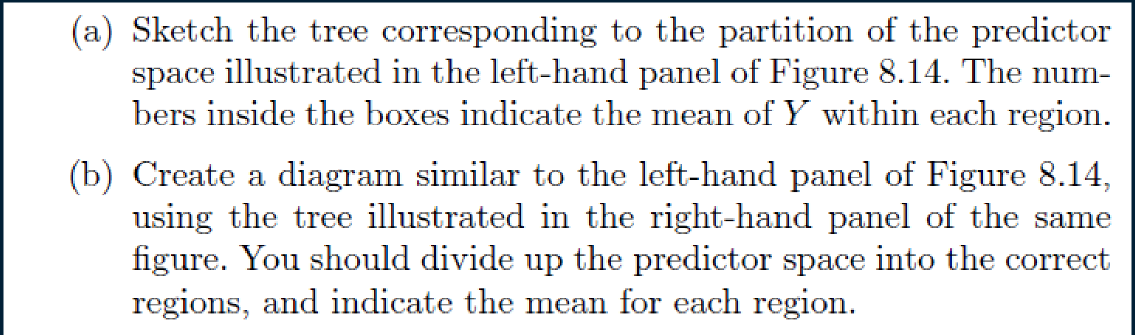
4. Suppose we fit a curve with basis functions $b_1(X) = I(0 \leq X \leq 2) - (X-1)I(1 \leq X \leq 2)$, $b_2(X) = (X-3)I(3 \leq X \leq 4) + I(4 < X \leq 5)$. We fit the linear regression model

$$Y = \beta_0 + \beta_1 b_1(X) + \beta_2 b_2(X) + \epsilon,$$

and obtain coefficient estimates $\hat{\beta}_0 = 1, \hat{\beta}_1 = 1, \hat{\beta}_2 = 3$. Sketch the estimated curve between $X = -2$ and $X = 6$. Note the intercepts, slopes, and other relevant information.

Sample Questions – Module 9

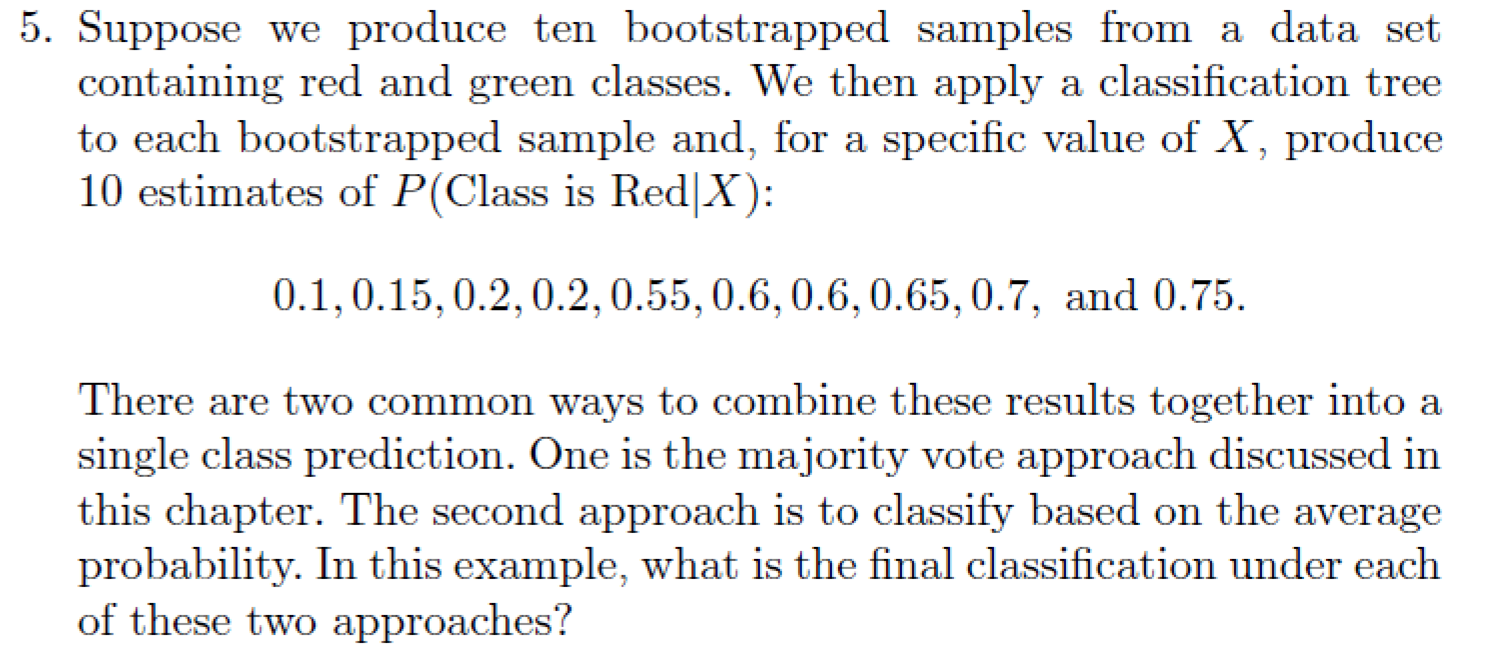




(a) Sketch the tree corresponding to the partition of the predictor space illustrated in the left-hand panel of Figure 8.14. The numbers inside the boxes indicate the mean of Y within each region.

(b) Create a diagram similar to the left-hand panel of Figure 8.14, using the tree illustrated in the right-hand panel of the same figure. You should divide up the predictor space into the correct regions, and indicate the mean for each region.

Sample Questions – Module 9



5. Suppose we produce ten bootstrapped samples from a data set containing red and green classes. We then apply a classification tree to each bootstrapped sample and, for a specific value of X, produce 10 estimates of $P(\text{Class is Red}|X)$:

$$0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, \text{ and } 0.75.$$

There are two common ways to combine these results together into a single class prediction. One is the majority vote approach discussed in this chapter. The second approach is to classify based on the average probability. In this example, what is the final classification under each of these two approaches?