CME 250:
Introduction to Machine Learning

Lecture 2:
Linear and Logistic Regression

Sherrie Wang
sherwang@stanford.edu

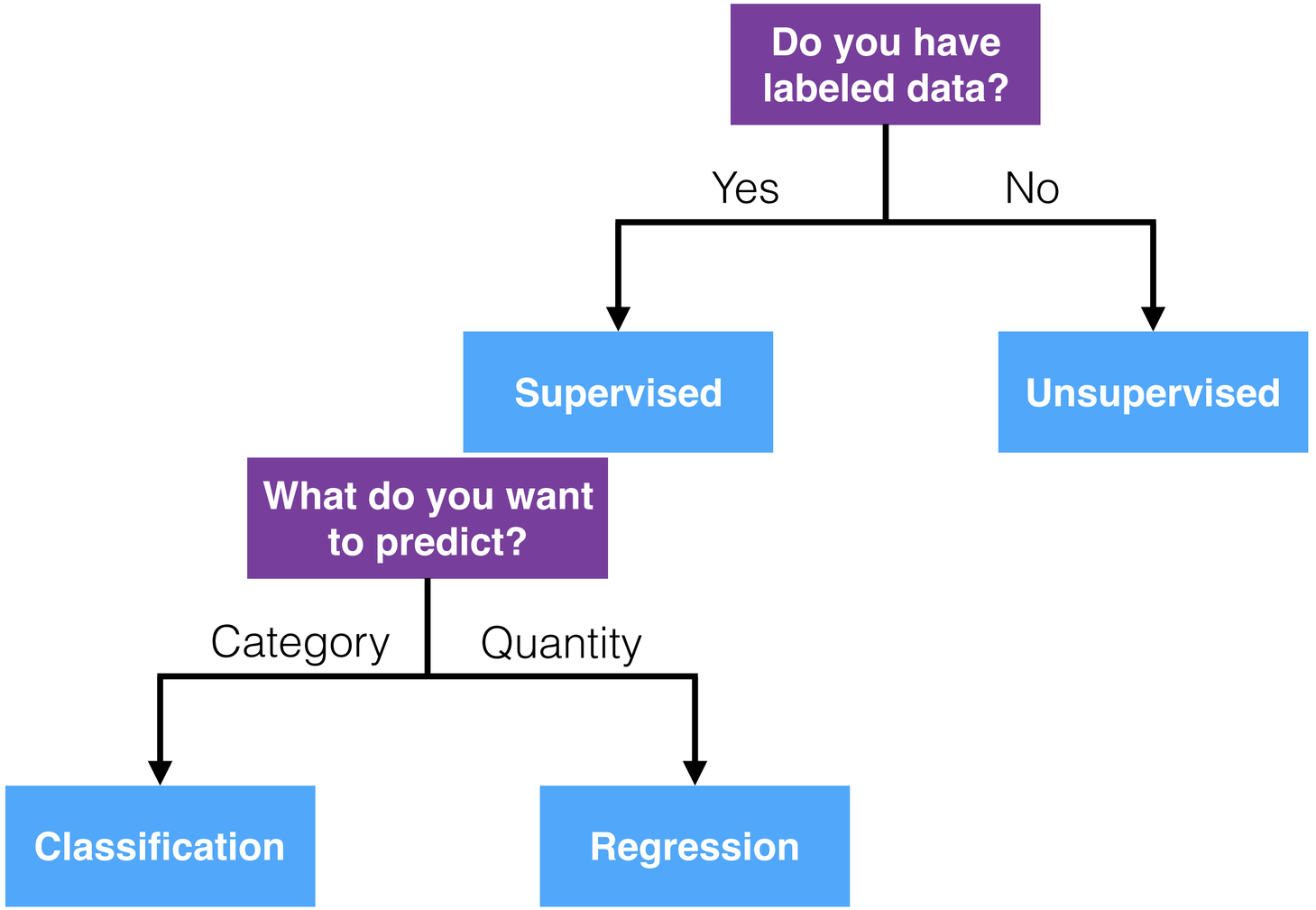
ICME

Agenda

Slides are online at cme250.stanford.edu

- Bias-variance trade-off
- Linear regression
  - Simple linear regression
  - What is a “good” fit?
  - Multiple linear regression
  - Variations on linear regression
- Logistic regression
  - What is a “good” fit?

Recall: Types of Machine Learning



Bias and Variance

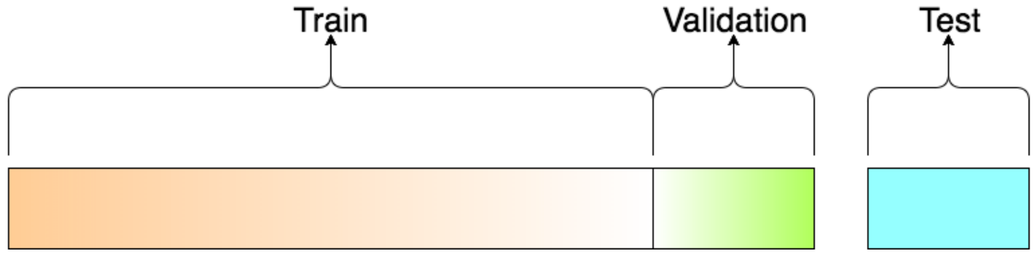
Assessing Model Performance

There are a number of metrics used to assess model performance on supervised tasks (regression and classification).

Key point: We want to know how good predictions are when we apply our method to previously unseen data.

Why? The ability to generalize to unseen data is what makes these methods useful.

Datasets



Training data

- Observations used to learn the model

Validation data

- Observations used to estimate error for parameter-tuning or model selection

Test data

- Observations used to measure performance on unseen data (how well the model generalizes)
- Not available to the algorithm during any part of the learning process

Assessing Model Performance

We want a method that gives high test set performance, or low test error.

What about high training set performance / low training set error?

There is no guarantee that the method with the highest training performance will have the highest test performance.

Model Flexibility

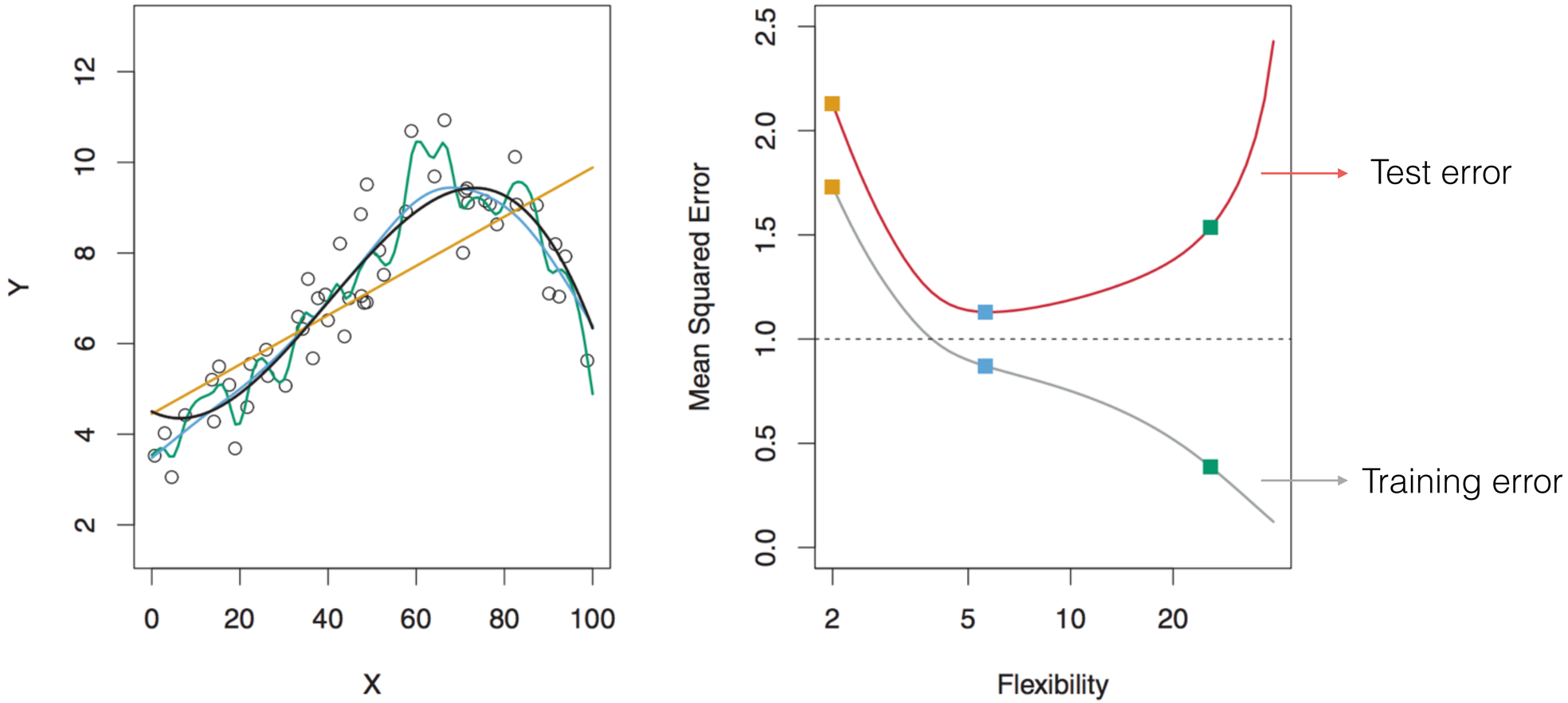


FIGURE 2.9, ISL (8th printing 2017)

Model Flexibility

As model flexibility increases, training error will decrease.

The model can fit more and more of the variance in the training set. Some of this variance, however, may be noise.

Therefore the test error may or may not decrease.

If training error is much larger than test error, the model is *overfitting*.

Model Flexibility

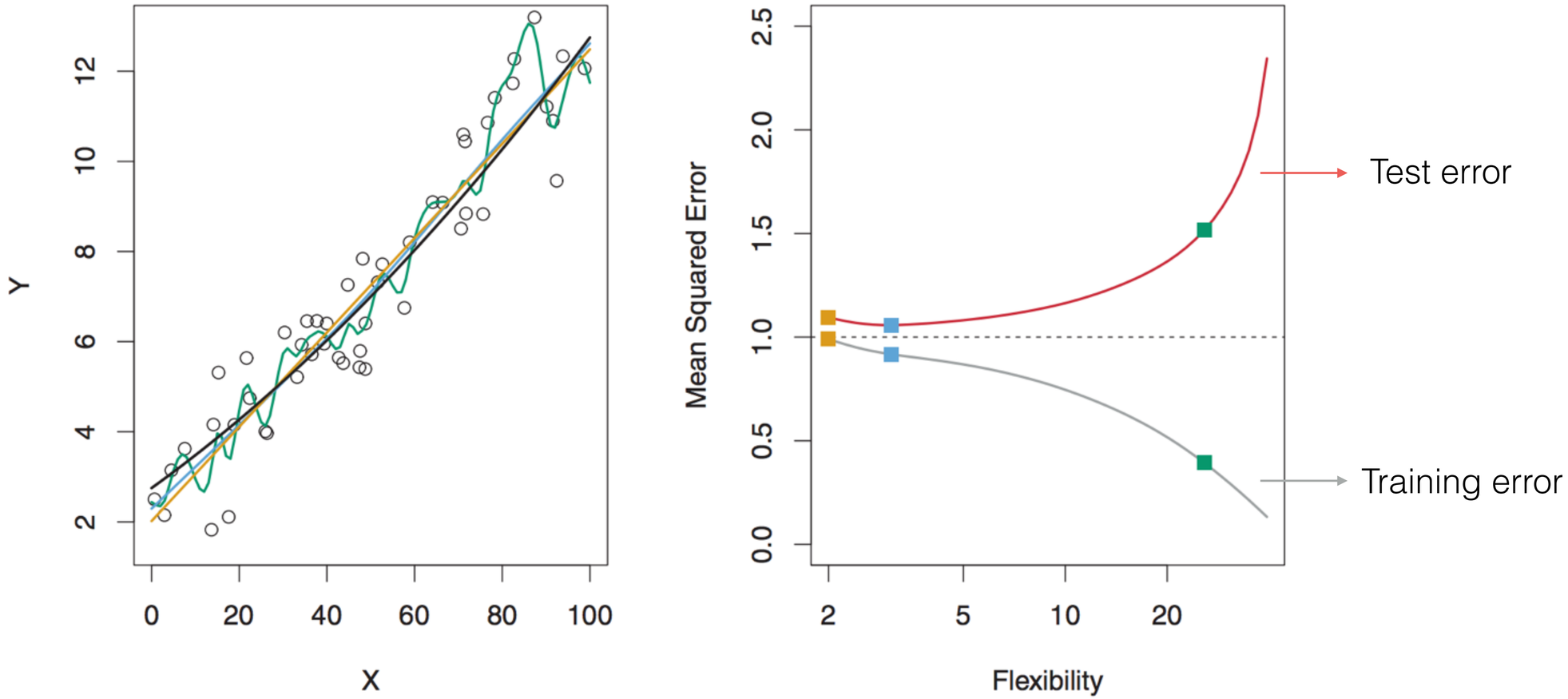


FIGURE 2.10, ISL (8th printing 2017)

Bias and Variance

Expected test error = Variance + Bias^2 + another term

$$\mathrm{E}\left[\left(y - \hat{f}\left(x\right)\right)^2\right] = \left(\mathrm{Bias}\left[\hat{f}\left(x\right)\right]\right)^2 + \mathrm{Var}\left[\hat{f}\left(x\right)\right] + \sigma^2$$

Bias = error caused by simplifying assumptions built into the model

Variance = how much the learned function will change if trained on a different training set

Bias-Variance Trade-off

Generally, more flexible methods have more variance and less bias.

Less flexible methods have more bias and less variance.

The best method for a task will balance the two types of error to achieve the lowest test error.

Bias-Variance Trade-off

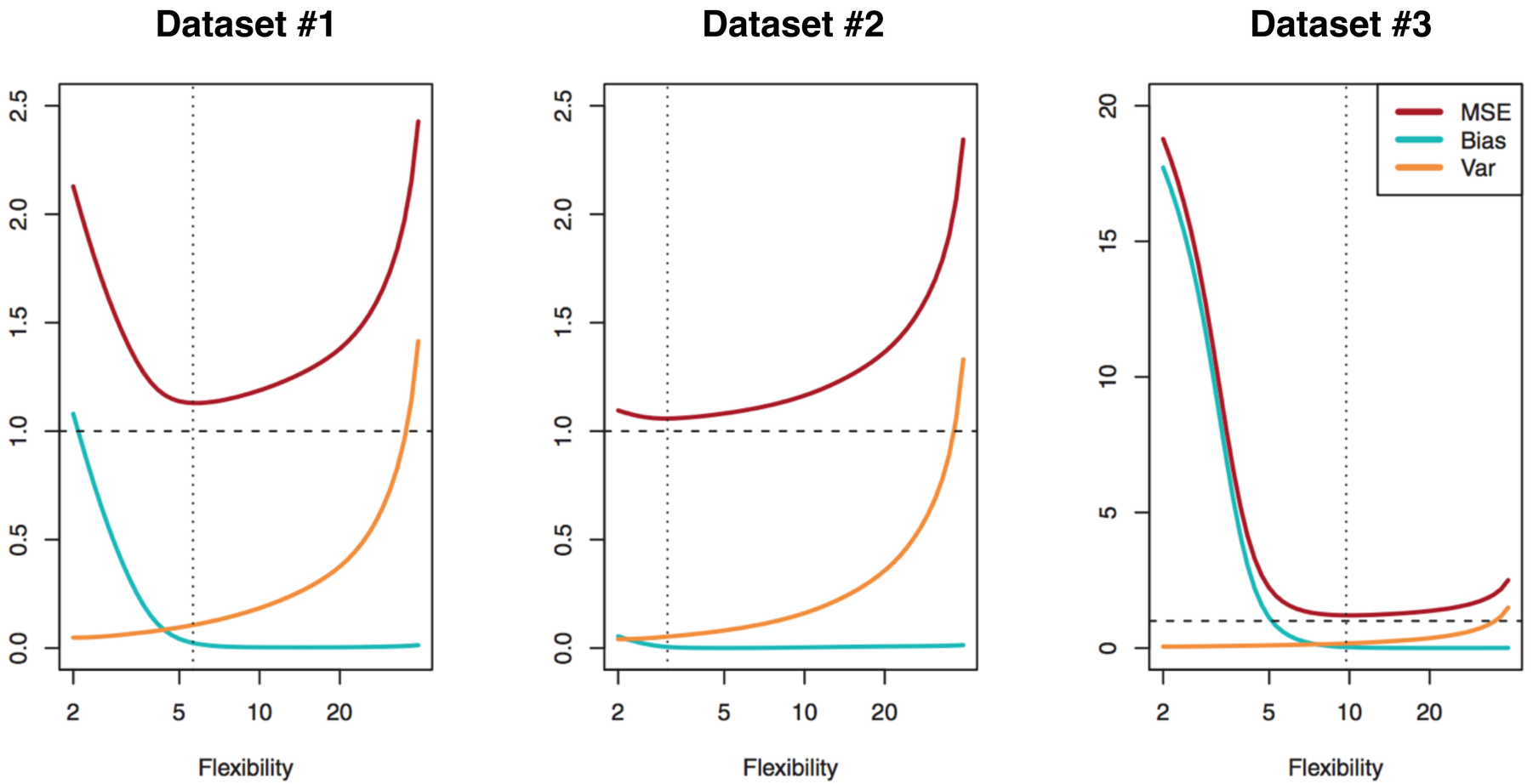


FIGURE 2.12, ISL (8th printing 2017)

Supervised Algorithm #2:
Linear Regression

Linear Regression

Simple supervised learning method, used to predict quantitative output values.

- Many machine learning methods are generalizations of linear regression.
- Illustrates key concepts in supervised learning while maintaining interpretability.

Simple Linear Regression

Predict a quantitative response Y on the basis of a single predictor variable X.

Assumes there is an approximately linear relationship between X and Y.

$$Y = \beta_0 + \beta_1 X + \epsilon$$

β_0 and β_1 are the *coefficients* or *parameters* of the linear model. In this case they represent the intercept and slope terms of a line.

Simple Linear Regression

Estimate βs using training data: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots (x^{(n)}, y^{(n)})$

Once βs are estimated, we denote them using "hats".

For a particular realization of X, aka $X = x$, the predicted output is denoted "y-hat":

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

Goal: Pick $\hat{\beta}_0, \hat{\beta}_1$ such that the model is a good fit to the training data.

$$y^{(i)} \approx \hat{\beta}_0 + \hat{\beta}_1 x^{(i)}, \quad i = 1, \ldots, n$$

Simple Linear Regression

Advertising dataset

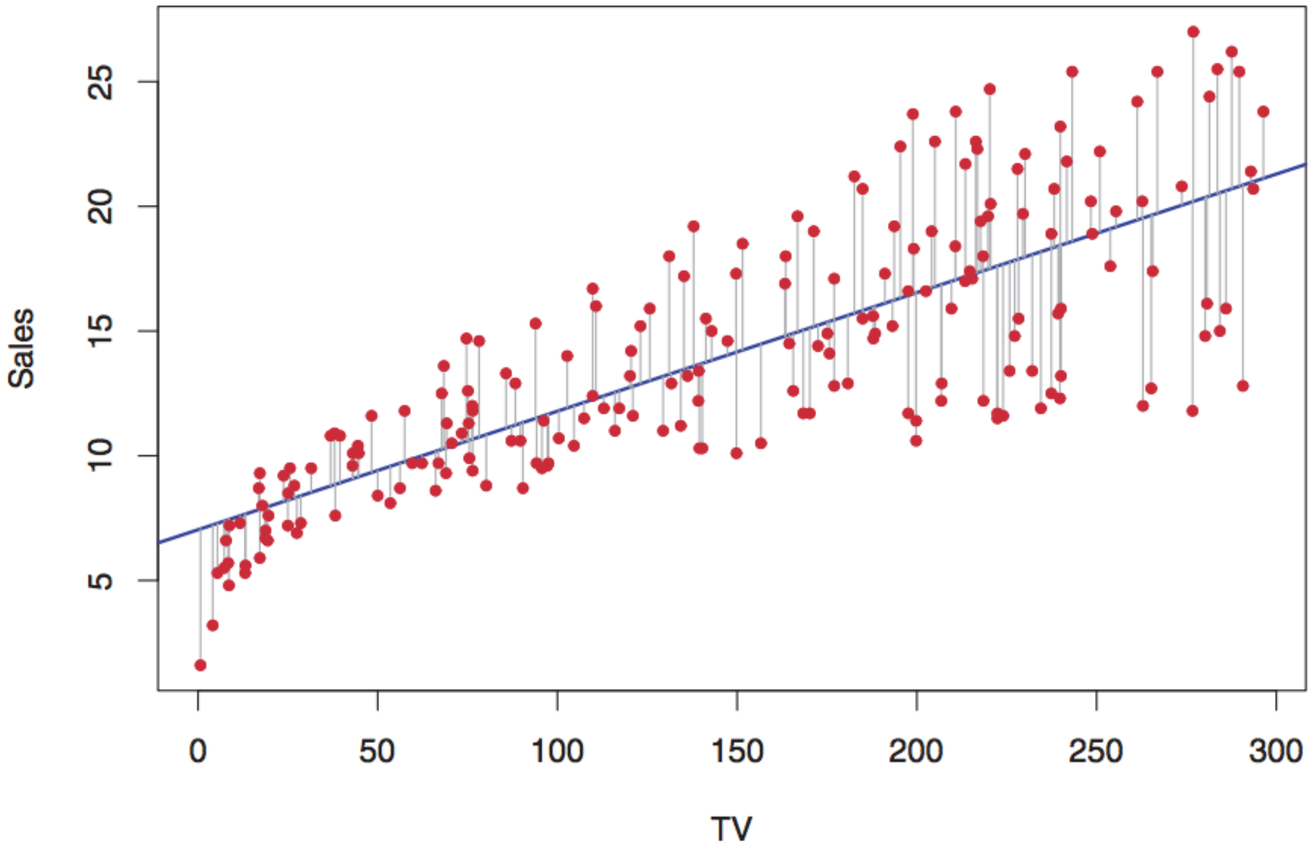


FIGURE 3.1, ISL (8th printing 2017)

Simple Linear Regression

Two related questions:

- How do we estimate the coefficients? (aka “fit the model”)
- What is a “good fit” to the data?

Least Squares

Typically, how well a linear model is fit to the data is measured using *least squares*.

Residual for the i-th sample:

$$\epsilon^{(i)} = y^{(i)} - \hat{y}^{(i)}$$

Residual sum of squares (RSS):

$$\text{RSS} = \epsilon^{(1)2} + \epsilon^{(2)2} + \ldots + \epsilon^{(n)2} = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2$$

Least Squares

The model fit using least squares finds $\hat{\beta}_0, \hat{\beta}_1$ that minimize the RSS.

$$\hat{\beta}_0, \hat{\beta}_1 = \arg\min_{\beta_0, \beta_1} \left[\sum_{i=1}^{n} \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)} \right) \right)^2 \right]$$

Recall that the extrema of a function can be found by setting its derivative to zero, and verified to be minima via the second derivative.

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} \left(x^{(i)} - \bar{x} \right) \left(y^{(i)} - \bar{y} \right)}{\sum_{i=1}^{n} \left(x^{(i)} - \bar{x} \right)^2}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

sample means

Least Squares in Pictures

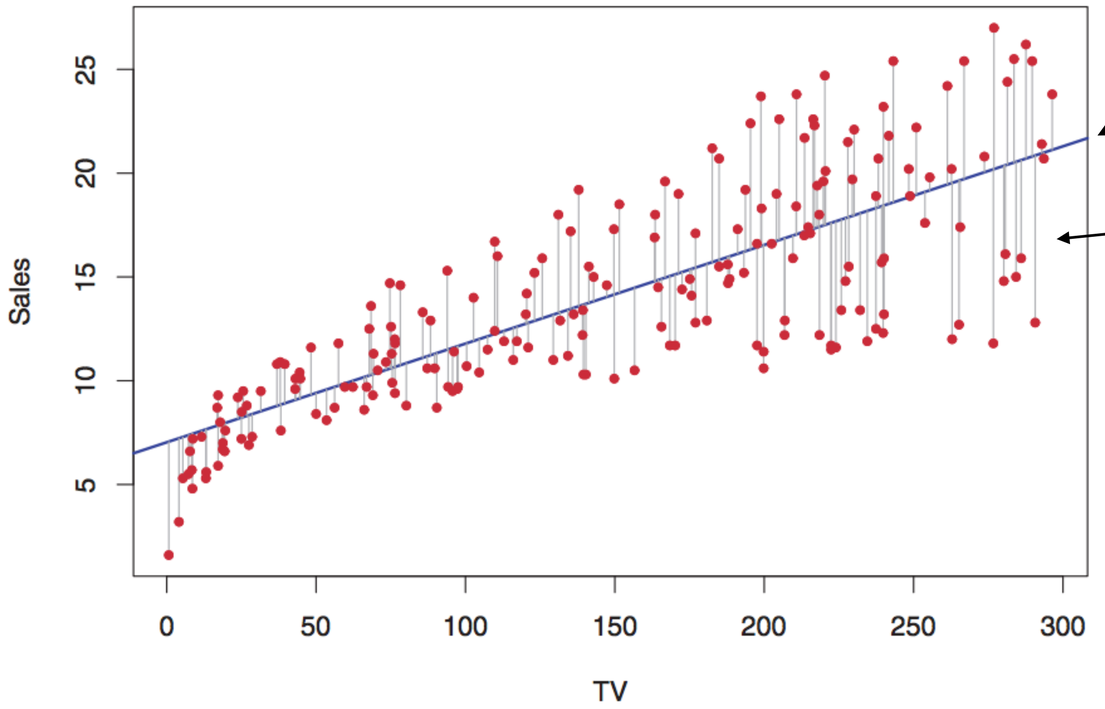


FIGURE 3.1, ISL (8th printing 2017)

$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$

RSS is the sum of the squares of all vertical gray lines.

As we vary the βs, RSS changes. Least squares finds βs that minimize RSS.

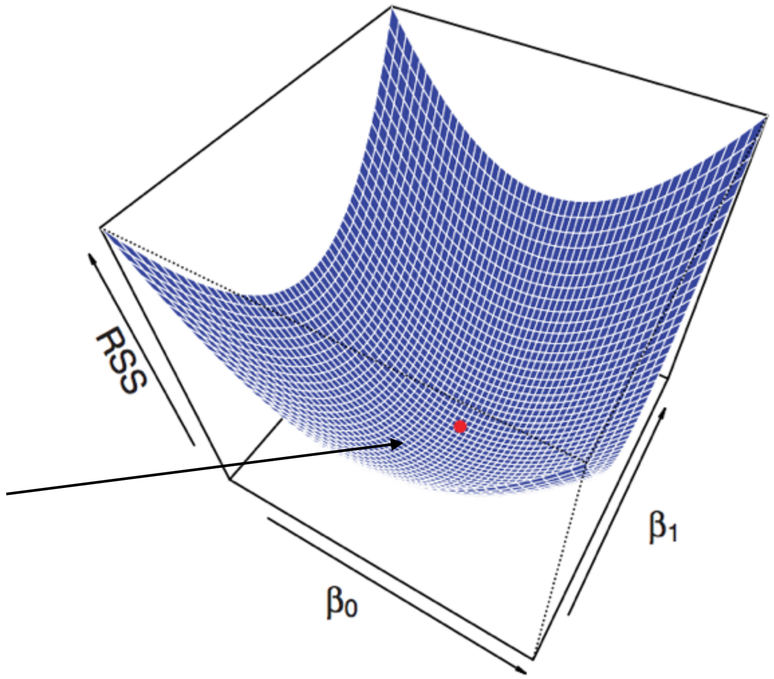


FIGURE 3.2, ISL (8th printing 2017)

How good is the model fit?

Linear regression is typically assessed using two related metrics:

- *Residual standard error* (RSE)

$$\mathrm{RSE} = \sqrt{\frac{1}{n-2}\mathrm{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2}$$

higher RSE means worse fit

How good is the model fit?

Linear regression is typically assessed using two related metrics:

- *Residual standard error* (RSE)

$$\mathrm{RSE} = \sqrt{\frac{1}{n-2}\mathrm{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2}$$

higher RSE means worse fit

- R^2 statistic

$$R^2 = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

TSS is the "total sum of squares" $\sum_{i=1}^{n}(y^{(i)} - \bar{y})^2$

higher R^2 means better fit

R^2 measures the proportion of variability in Y explained by X.

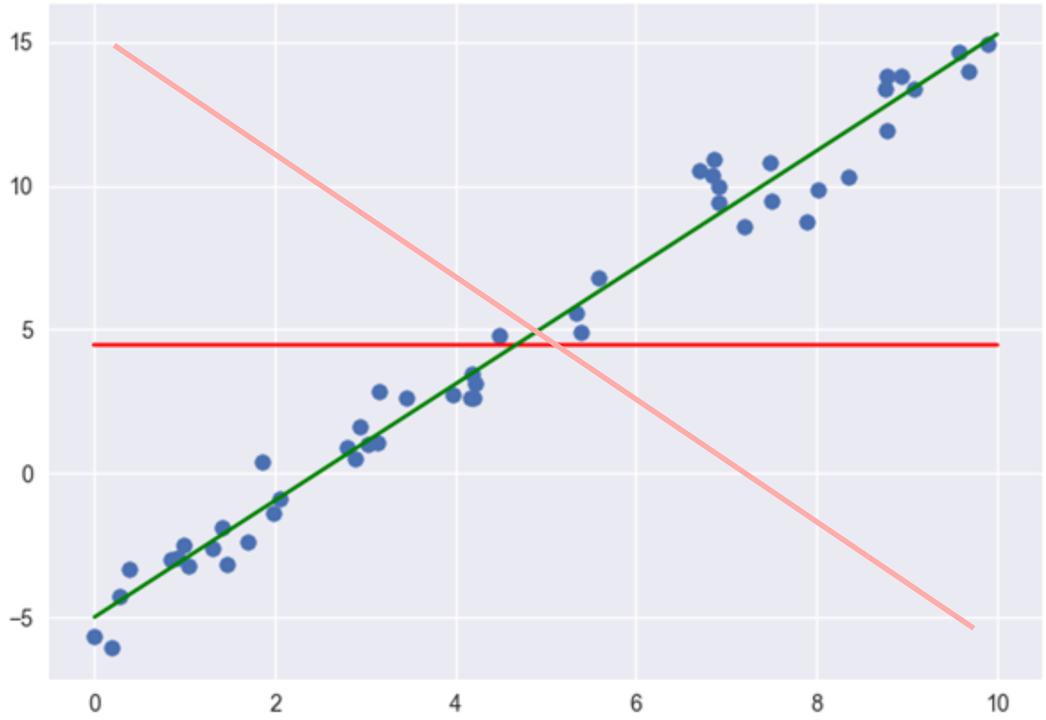
What range of values can R^2 take?

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

$$\text{RSS} = \epsilon^{(1)2} + \epsilon^{(2)2} + \ldots + \epsilon^{(n)2} = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2$$

$$\text{TSS} = \sum_{i=1}^{n} (y^{(i)} - \bar{y})^2$$

What range of values can R^2 take?



How good is the model fit?

What values of RSE and R^2 are "good"?

Depends on the domain and application.

- In physics, we may know the data comes from a linear model. In such a case, we'd require R^2 close to 1 in order to call the fit good.

- In biology, social sciences, and other domains, a linear model may be a crude approximation. Existing models may also not be good at estimating Y, so a model with R^2 of 0.4 may be considered good.

Multiple Linear Regression

What if our dataset contains multiple input dimensions X_j?

We could run a simple linear regression for each input dimension.

Is this satisfactory?

Multiple Linear Regression

Predict the response variable using more than one predictor variable.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

Here, X_j represents the j-th predictor.

We interpret β_j as the average effect on Y of a 1 unit increase in X_j, holding all other predictors fixed.

Multiple Linear Regression

3 Simple Regressions

	Coefficient	Std. error	t-statistic	p-value
Intercept	7.0325	0.4578	15.36	< 0.0001
TV	0.0475	0.0027	17.67	< 0.0001

	Coefficient	Std. error	t-statistic	p-value
Intercept	9.312	0.563	16.54	< 0.0001
radio	0.203	0.020	9.92	< 0.0001

	Coefficient	Std. error	t-statistic	p-value
Intercept	12.351	0.621	19.88	< 0.0001
newspaper	0.055	0.017	3.30	0.00115

1 Multiple Regression

	Coefficient	Std. error	t-statistic	p-value
Intercept	2.939	0.3119	9.42	< 0.0001
TV	0.046	0.0014	32.81	< 0.0001
radio	0.189	0.0086	21.89	< 0.0001
newspaper	-0.001	0.0059	-0.18	0.8599

Multiple Linear Regression

3 Simple Regressions

	Coefficient	Std. error	t-statistic	p-value
Intercept	7.0325	0.4578	15.36	< 0.0001
TV	0.0475	0.0027	17.67	< 0.0001

	Coefficient	Std. error	t-statistic	p-value
Intercept	9.312	0.563	16.54	< 0.0001
radio	0.203	0.020	9.92	< 0.0001

	Coefficient	Std. error	t-statistic	p-value
Intercept	12.351	0.621	19.88	< 0.0001
newspaper	0.055	0.017	3.30	0.00115

1 Multiple Regression

	Coefficient	Std. error	t-statistic	p-value
Intercept	2.939	0.3119	9.42	< 0.0001
TV	0.046	0.0014	32.81	< 0.0001
radio	0.189	0.0086	21.89	< 0.0001
newspaper	-0.001	0.0059	-0.18	0.8599

Variable Correlations

	TV	radio	newspaper	sales
TV	1.0000	0.0548	0.0567	0.7822
radio		1.0000	0.3541	0.5762
newspaper			1.0000	0.2283
sales				1.0000

Multiple Linear Regression

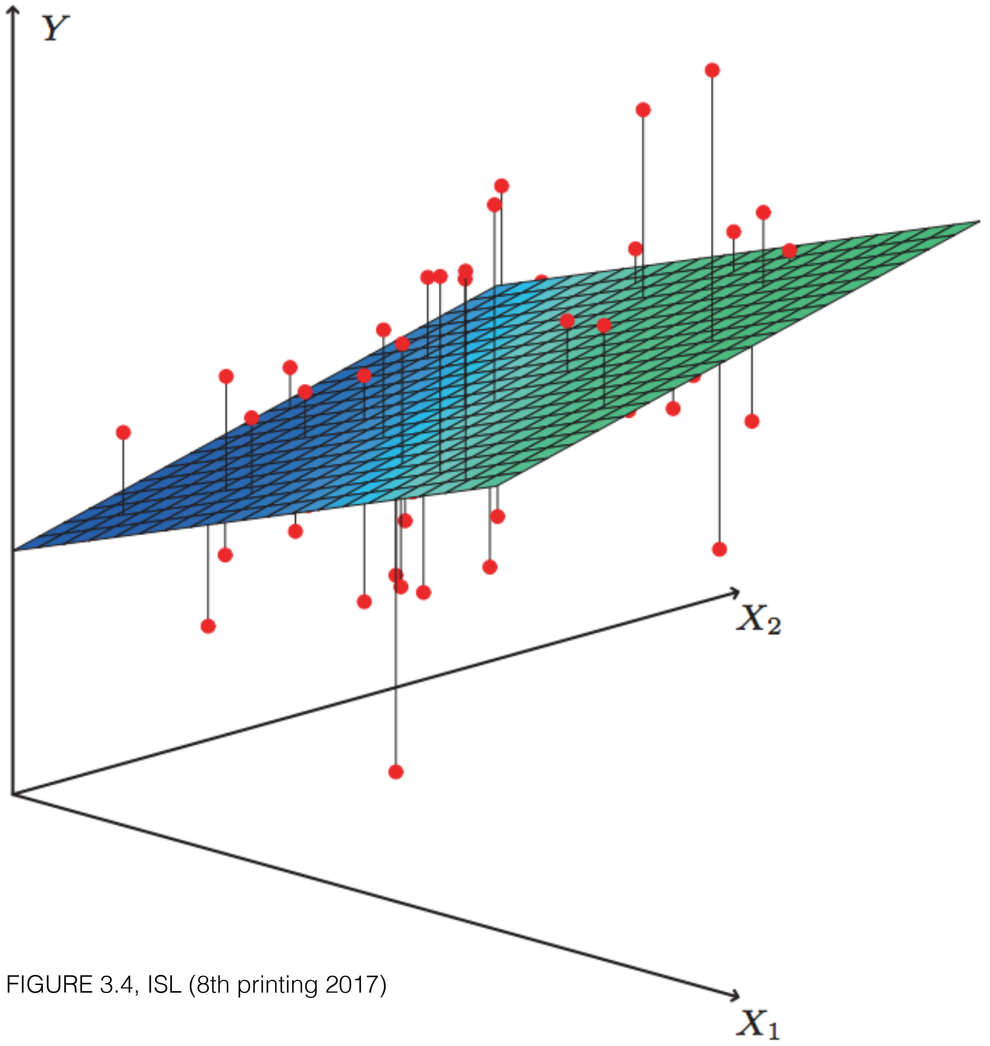


FIGURE 3.4, ISL (8th printing 2017)

Multiple Linear Regression

Another way to write

$$y^{(i)} = \beta_0 + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \ldots + \beta_p x_p^{(i)} + \epsilon^{(i)}$$

is to use matrix notation.

Multiple Linear Regression

Define:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_p^{(1)} \\ 1 & x_1^{(2)} & \dots & x_p^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & \dots & x_p^{(n)} \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} \quad \hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_p \end{bmatrix}$$

Then:

$$\mathbf{Y} = \mathbf{X}\vec{\beta} + \vec{\epsilon}$$

Least Squares

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_p \end{bmatrix}$$

The model fit using least squares finds $\hat{\beta}$ that minimize the RSS.

$$\hat{\beta} = \arg\min_{\vec{\beta}} ||\mathbf{Y} - \mathbf{X}\vec{\beta}||_2^2$$

Notation:
square of L2-norm
of the vector inside ||

The *normal equations* give us the analytical solution for $\hat{\beta}$.

$$\mathbf{X}^\top \mathbf{X}\hat{\beta} = \mathbf{X}^\top \mathbf{Y} \longrightarrow \hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y}$$

How good is the model fit?

Multiple linear regression can also be assessed using the 2 metrics previously discussed.

- *Residual standard error* (RSE)
- R^2 statistic

$$\mathrm{RSE} = \sqrt{\frac{1}{n-p-1}\mathrm{RSS}} = \sqrt{\frac{1}{n-p-1}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2}$$

$$R^2 = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

Qualitative Inputs

How do we include qualitative inputs in regression?

- E.g. predictor is “gender”: “male” or “female”

If only two possible values, we can create a *dummy variable*

$$X_j = \begin{cases} 1 & \text{if female} \\ 0 & \text{if male} \end{cases}$$

Qualitative Inputs

If more than two possible values, introduce multiple dummy variables

- E.g. predictor is “eye color”: “blue”, “green”, or “brown”

$$X_j = \begin{cases} 1 & \text{if blue} \\ 0 & \text{if not blue} \end{cases}$$

$$X_{j+1} = \begin{cases} 1 & \text{if brown} \\ 0 & \text{if not brown} \end{cases}$$

Potential Problems

- Non-linearity of response-predictor relationships
- Non-additivity of predictors
- Non-constant variance of error terms
- Outliers can mislead model performance
- High-leverage points overly influence βs
- Correlation of error terms
- Collinearity

Non-linearity of Data

Residual plots can help diagnose if this is an issue.

Plot residuals vs. fitted values.

If there is a pattern in the residual plot, then the linearity assumption is suspect.

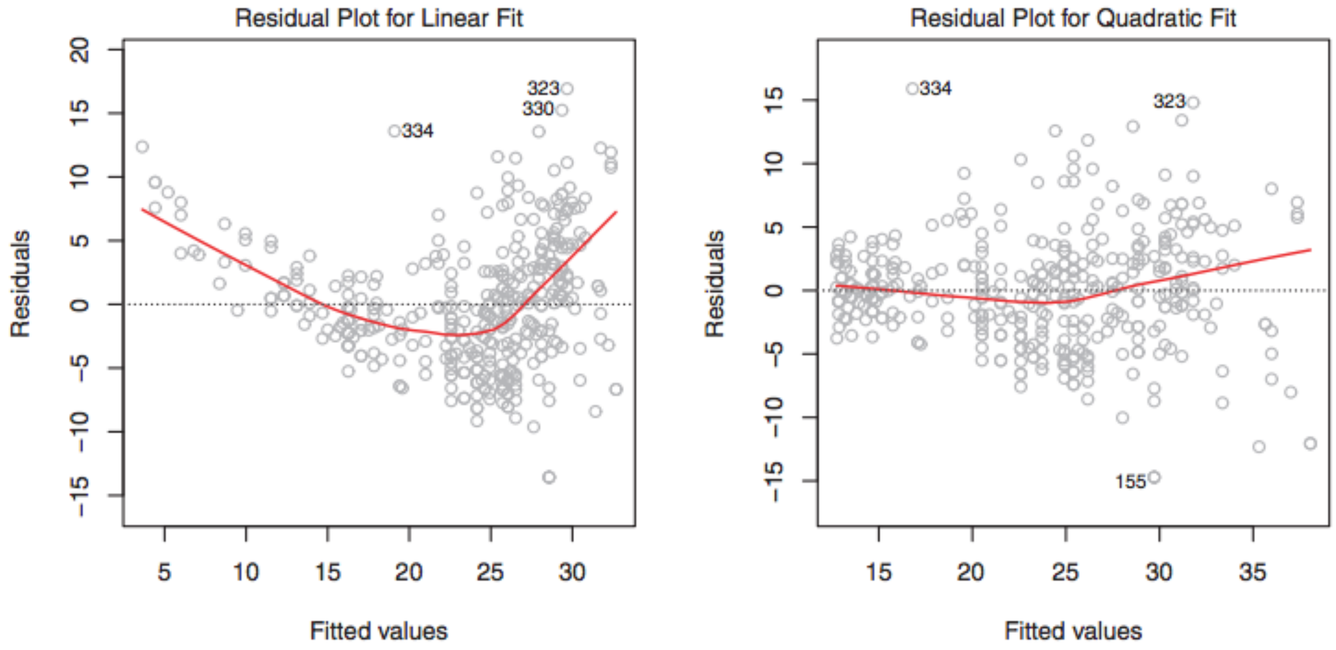


FIGURE 3.11, ISL (8th printing 2017)

Non-linearity of Data

A simple way to extend linear regression to model non-linear relationships is via *polynomial regression*.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \epsilon$$

This is still a linear model; it can be fit via least squares.

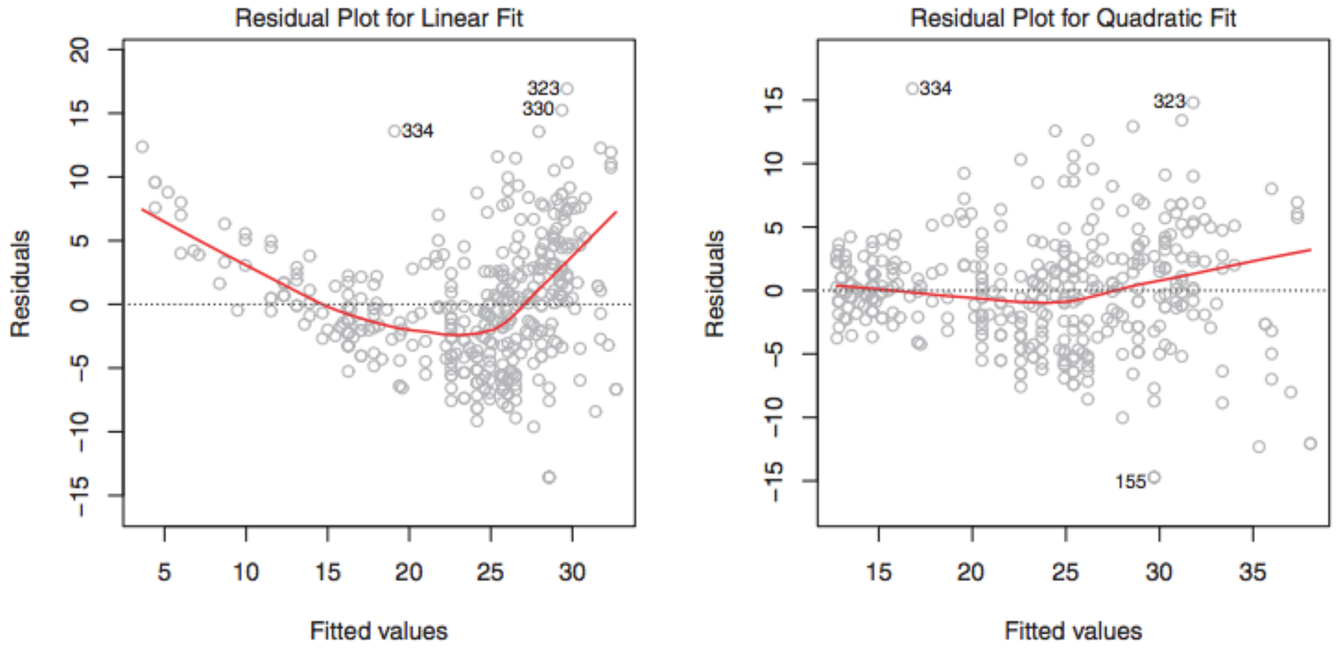


FIGURE 3.11, ISL (8th printing 2017)

Non-additivity of Predictors

Additivity means each predictor X_j affects Y independently of the value of other predictors.

If this is not true, we can extend linear regression by including *interaction effects*.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon$$

Non-constant Variance of Error

Can be seen in a funnel shape in the residual plot.

Standard errors, confidence intervals, and hypothesis tests rely on constant variance assumption.

One possible solution is to transform the response Y using a concave function (log, square root).

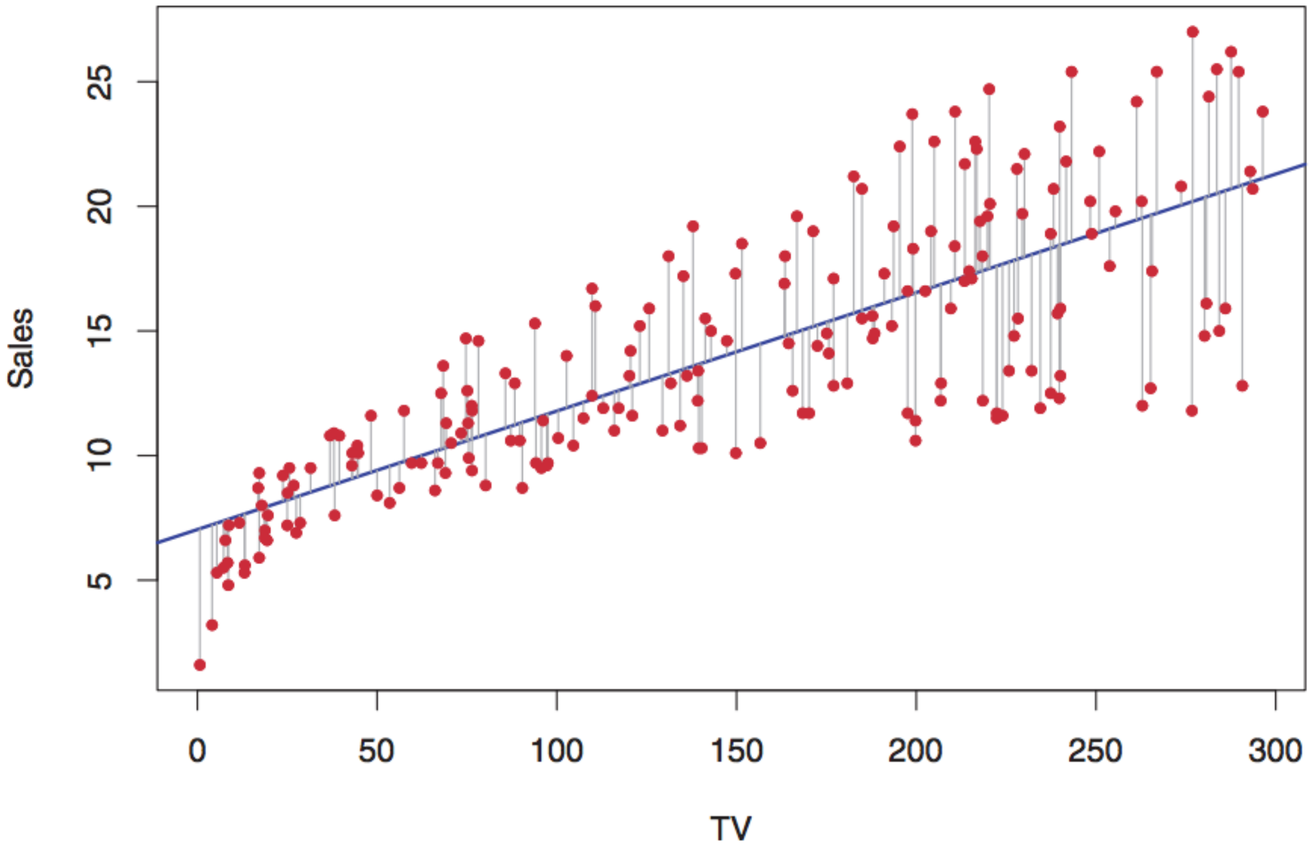
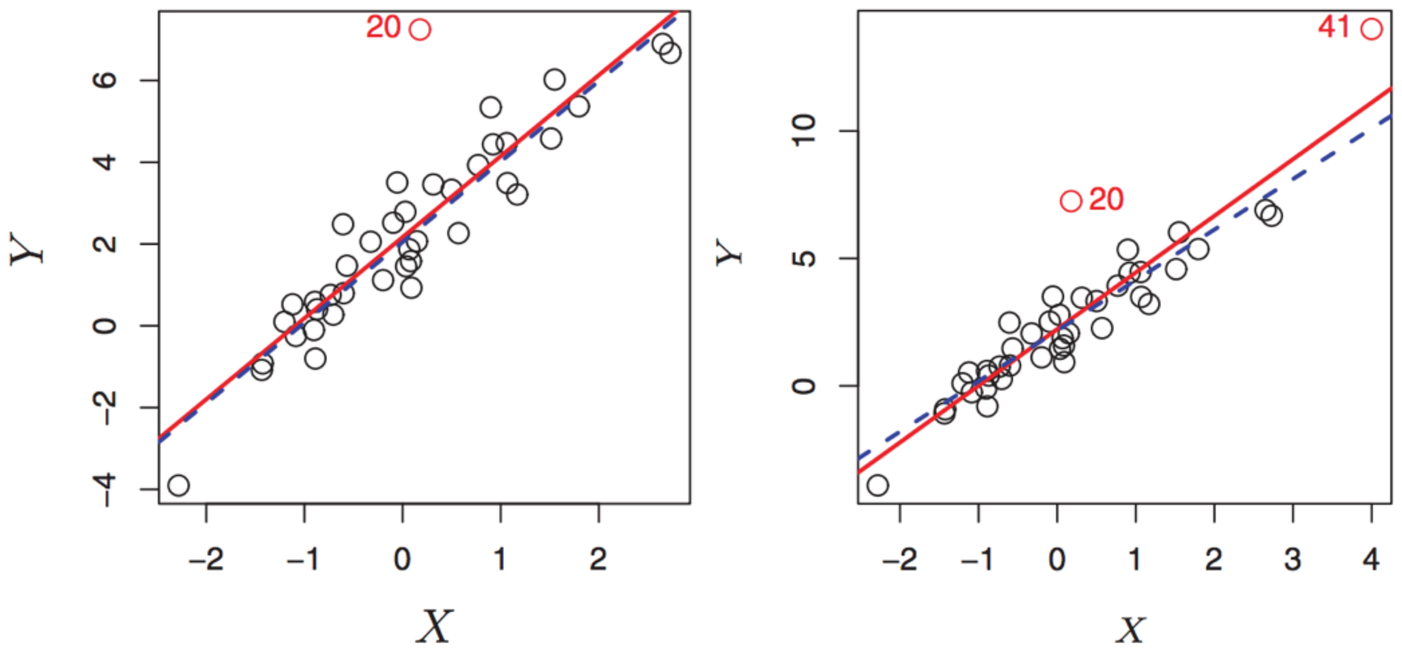


FIGURE 3.1, ISL (8th printing 2017)

Outliers and High-leverage Points

Outliers have unusual Y values, and can inflate RSE and deflate R^2.

High-leverage points have unusual values for predictors X, and influence least squares line substantially.



Other Questions

Not addressed in this course, but important to consider:

- Does the data contain evidence of a relationship between X and Y?
- Are the estimated coefficients close to the true ones?

Future lecture:

- If we have many predictors X_j, which one(s) do we include in the model?

Linear Regression Summary

Advantages:

- Simple model
- Interpretable coefficients
- Can obtain good results with small datasets

Disadvantages:

- Model may be too simple to make accurate predictions over large range of values
- Sensitive to outliers in data due to minimization of squared error

Linear Regression in `sklearn`

```
from sklearn.linear_model import LinearRegression


linreg = LinearRegression()
linreg.fit(X, y)
linreg.coef_          # coefficients of input
linreg.intercept_     # model intercept
y_hat = linreg.predict(y)
```

Assessment Metrics in `sklearn`

```
from sklearn.metrics import r2_score,
mean_squared_error


r2 = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
```

Supervised Algorithm #3:
Logistic Regression

Classification

Classification methods are used to predict **qualitative** output values.

- Assign each observation to a class / category
- e.g. K-nearest neighbor classifier from Lecture 1

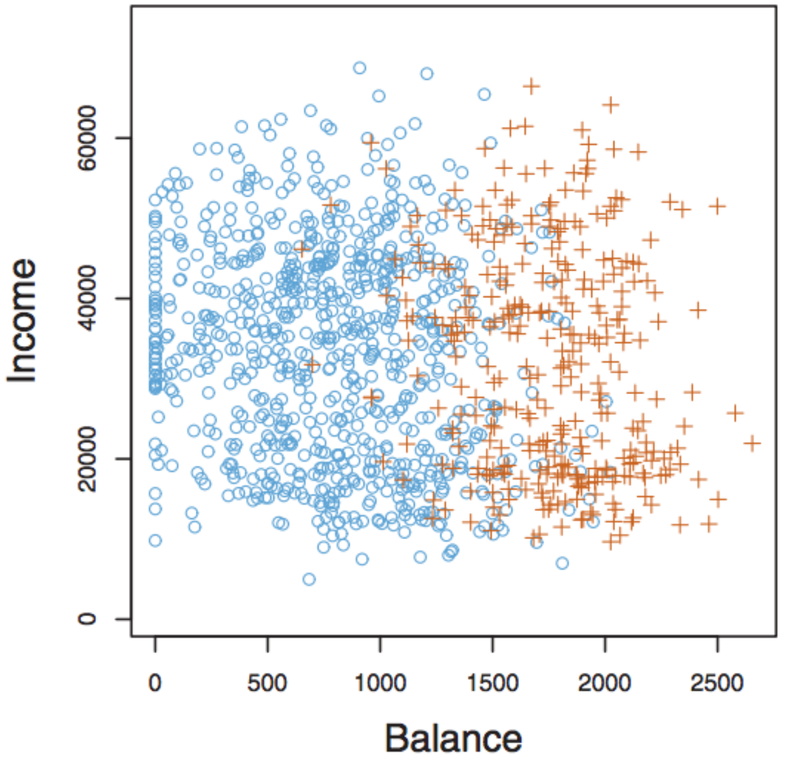


FIGURE 4.1, ISL (8th printing 2017)

Logistic Regression

Binary classification: Y takes on two values, 0 or 1, corresponding to 2 classes.

Logistic regression models binary classification as

$$Pr(Y \text{ belongs to class } 1 \mid X)$$

- Set threshold to obtain class decisions
- Extension of linear regression for probabilities in [0, 1]

Linear regression?

Why don’t we use this expression to model probability?

$$Pr(Y = 1|X) = \beta_0 + \beta_1 X$$

Logistic Regression

Logistic (sigmoid) function bounds output $Y \in (0, 1)$

Logistic function

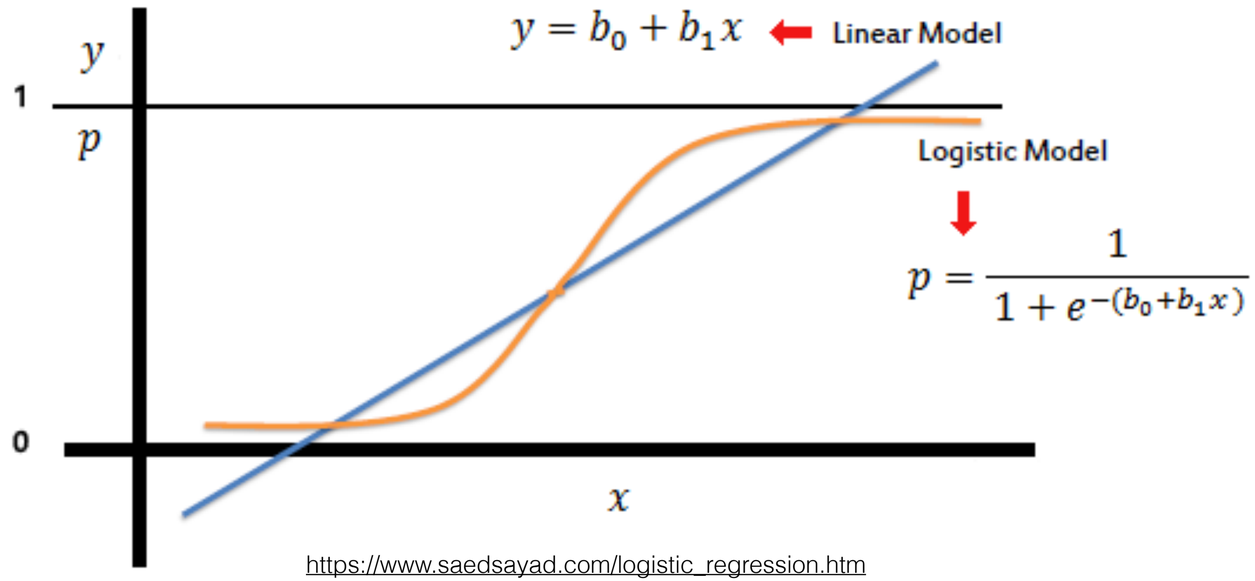
$$\sigma(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}$$

- S-shaped curve
- Always takes values in (0, 1), which are valid probabilities

Logistic regression model:

$$Pr(Y = 1|X) = \sigma(\beta_0 + \beta_1 X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

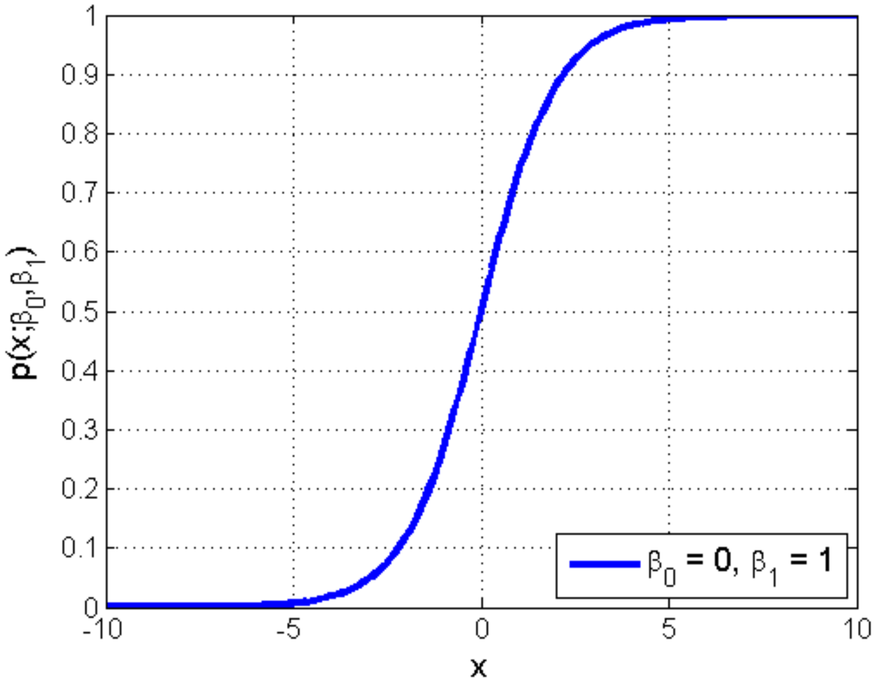
Logistic Regression



https://www.saedsayad.com/logistic_regression.htm

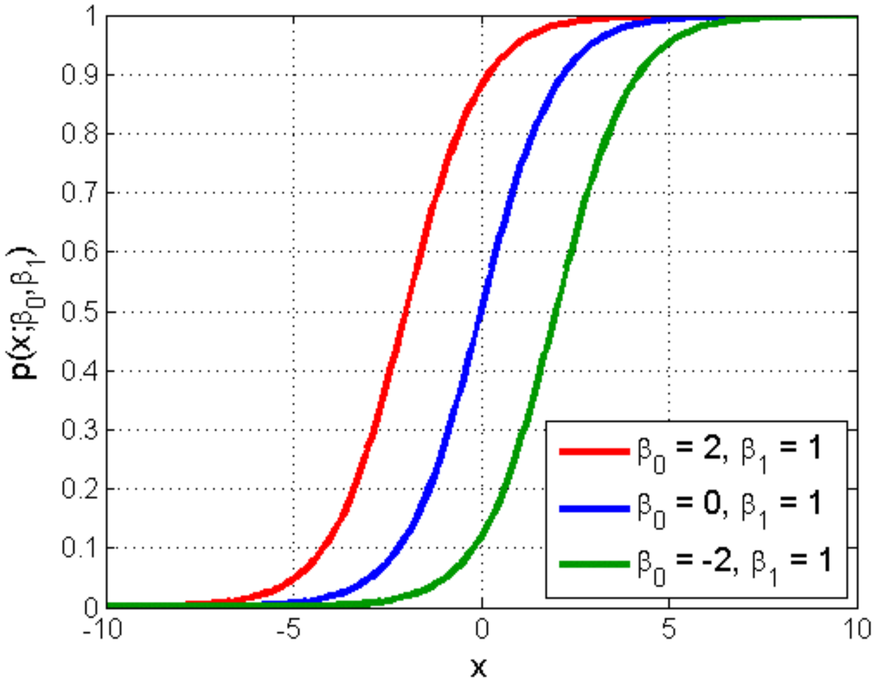
$$Pr(Y = 1|X) = \sigma(\beta_0 + \beta_1 X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

Logistic Regression



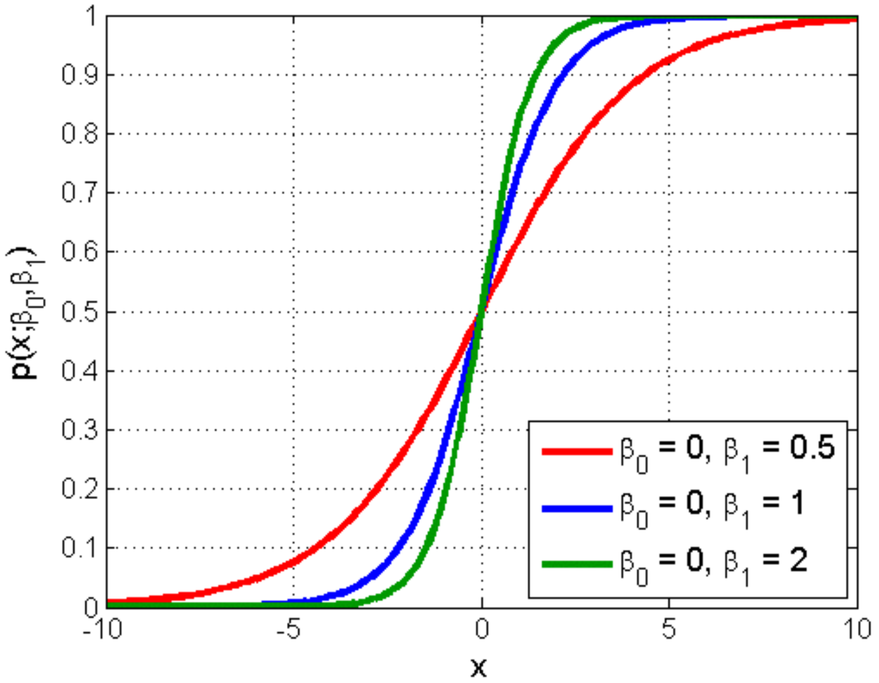
$$Pr(Y = 1|X) = \sigma(\beta_0 + \beta_1 X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

Logistic Regression



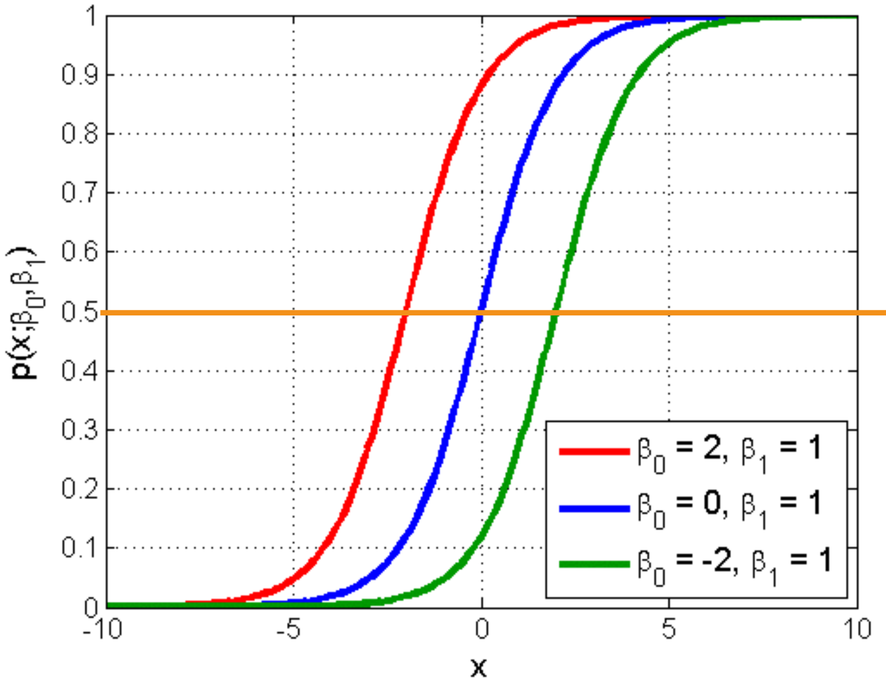
$$Pr(Y = 1|X) = \sigma(\beta_0 + \beta_1 X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

Logistic Regression



$$Pr(Y = 1|X) = \sigma(\beta_0 + \beta_1 X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

Logistic Regression



Threshold

$$Pr(Y = 1|X) = \sigma(\beta_0 + \beta_1 X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

Logistic Regression

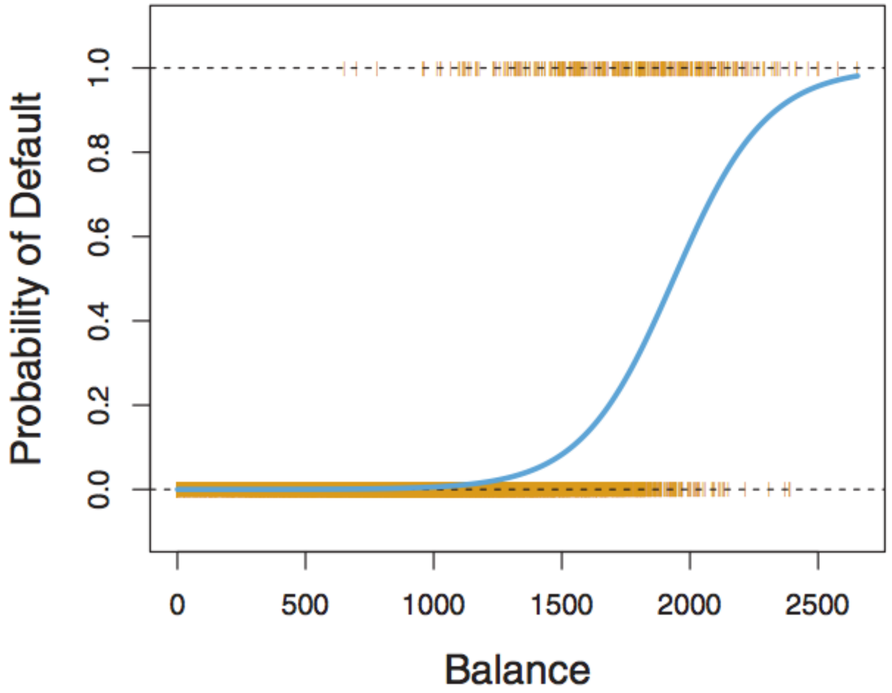


FIGURE 4.2, ISL (8th printing 2017)

$$Pr(Y = 1|X) = \sigma(\beta_0 + \beta_1 X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

Estimating Coefficients (β)

Recall that for linear regression, βs are estimated using least squares on the training data.

For logistic regression, there is no closed form solution for βs obtainable by taking the derivative and setting to zero.

Instead, βs are estimated using *maximum likelihood estimation*.

Multiple Logistic Regression

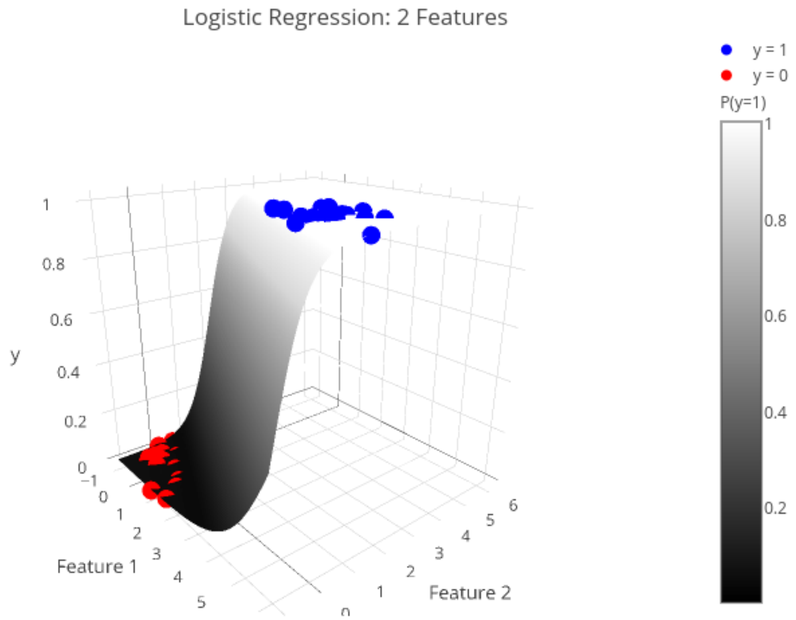
We can extend logistic regression to the case of multiple predictor variables.

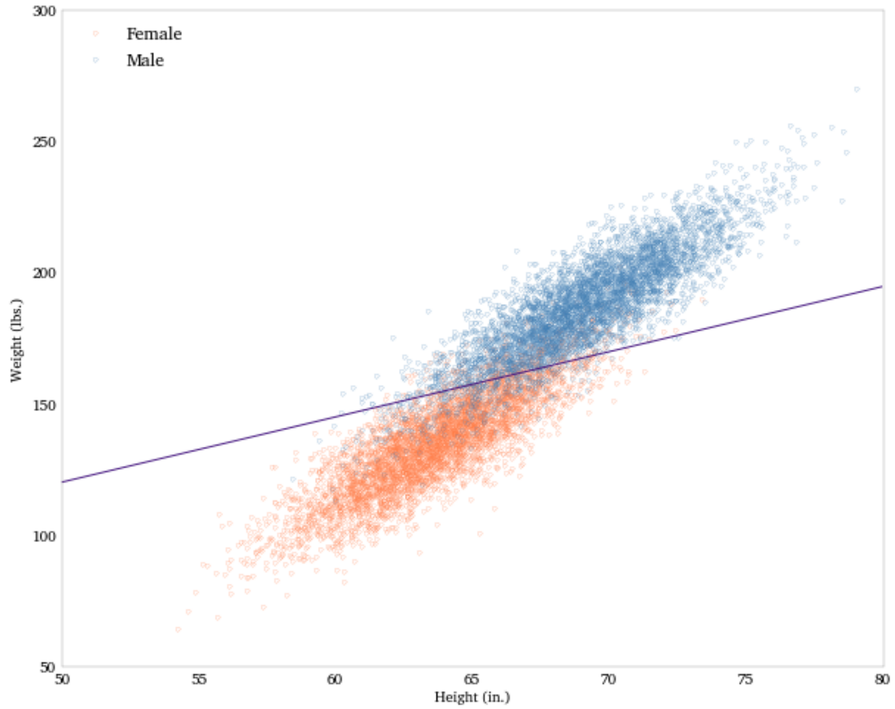
$$Pr(Y = 1|\mathbf{X}) = \sigma(\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p)$$

$$= \frac{e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}}$$

Multiple Logistic Regression

We can extend logistic regression to the case of multiple predictor variables.





https://florianhartl.com/logistic-regression-geometric-intuition.html

How good is the model fit?

Classifier performance can be summarized in a table called the *confusion matrix*.

- "Good performance" is when TP, TN large and FP, FN small
- Can be computed for training, validation, and test sets. Test set informs you about model generalizability.

		Predicted class	
		0	**1**
True class	**0**	True Positive (TP)	False Negative (FN)
	1	False Positive (FP)	True Negative (TN)

How good is the model fit?

Some common metrics to assess performance:

- Accuracy: $(\mathrm{TP} + \mathrm{TN}) / n$
- Recall: $\mathrm{TP} / (\mathrm{TP} + \mathrm{FN})$
- Precision: $\mathrm{TP} / (\mathrm{TP} + \mathrm{FP})$
- Specificity: $\mathrm{TN} / (\mathrm{FP} + \mathrm{TN})$
- False positive rate: $\mathrm{FP} / (\mathrm{FP} + \mathrm{TN})$

Predicted class

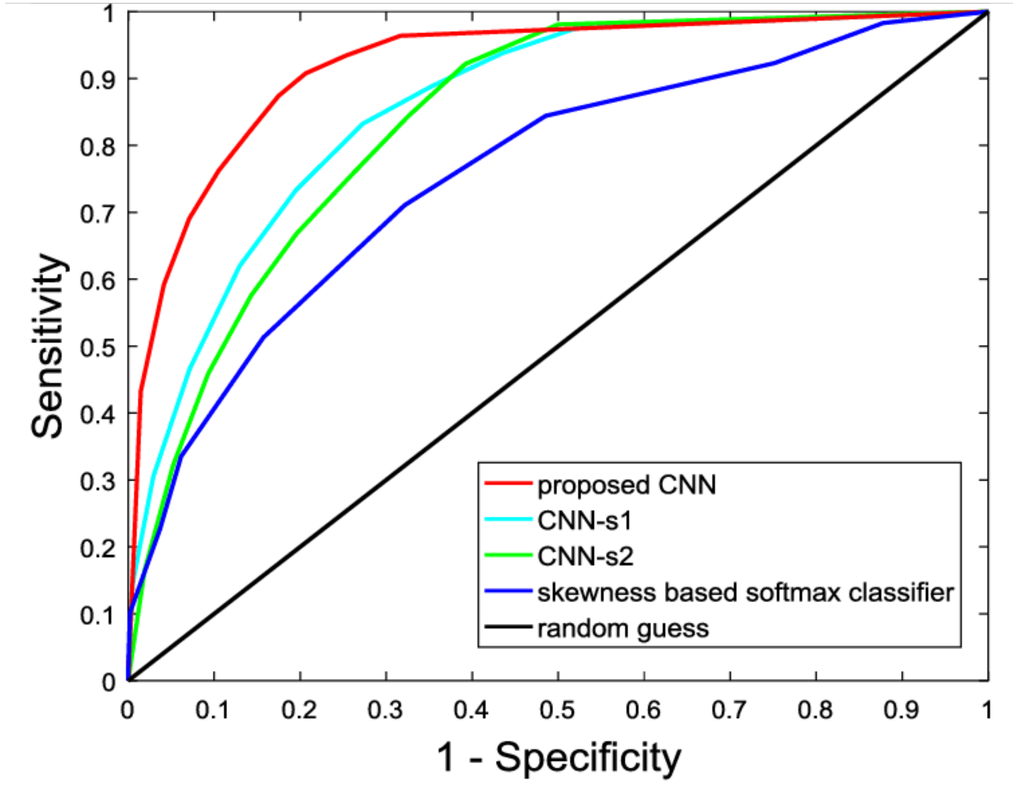
True class	**0**	**1**
0	True Positive (TP)	False Negative (FN)
1	False Positive (FP)	True Negative (TN)

How good is the model fit?

ROC (receiver operating characteristic) curve summarizes the trade-off between recall and false positive rate. Area under the curve summarizes in one metric.

Many classifiers have a “knob” or threshold that can be adjusted to make the classifier more or less conservative in predicting $Y = 1$.

Trade-off: more true positive (TP) —> more false positives (FP)



Logistic Regression Summary

Advantages:

- Extension of linear regression, simple
- Interpretable: log-odds are linear in predictors
- No hyperparameters to tune

Disadvantages:

- Cannot model complex decision boundaries

Dataset Splitting in `sklearn`

```
import pandas as pd
from sklearn.model_selection import train_test_split


data = pd.read_csv('dataset.csv')
X = data['input']
y = data['output']
X_train, X_test, y_train, y_test =
                train_test_split(X, y, test_size=0.2)
```

Logistic Regression in `sklearn`

```
from sklearn.linear_model import LogisticRegression


logreg = LogisticRegression()
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
```

Assessment Metrics in `sklearn`

```
from sklearn.metrics import accuracy_score,
precision_score, recall_score, confusion_matrix


accuracy = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
```