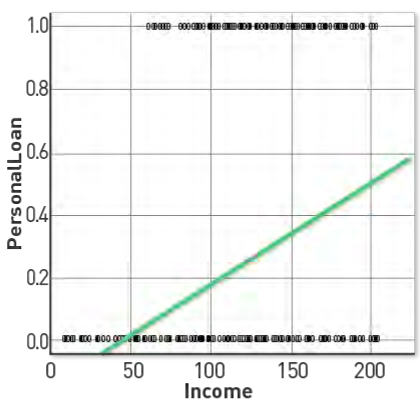
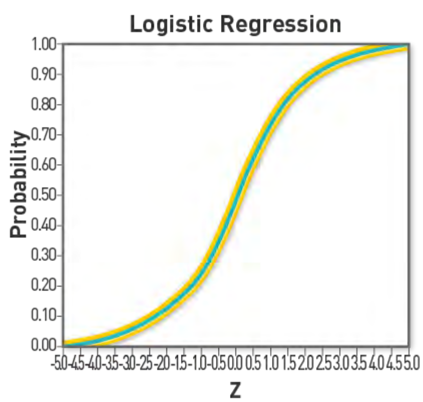
Choice Models

- Choice models examine binary behavior or choices.
  - E.g., a customer walks into a store and either buys a TV or doesn't.
  - The Y variable is not a continuous number but is binary (i.e., zero or one).
- Choice models apply to:
  - Purchases
  - Elections of government candidates
  - Loan and credit approvals
  - Anything where there is a choice of yes or no, zero or one.

A Problem with Choice Models





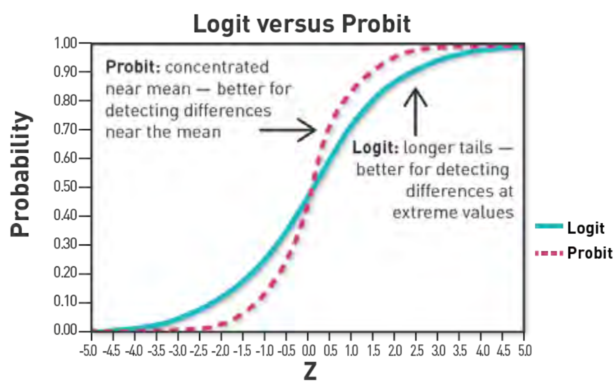
- Linear models don't fit.
- Logit (logistic) models do fit.

Logistic Regression and Logit

- Logistic regression assumes that the Y variable is a probability, ranging from zero to one.
- The regression determines the likelihood that someone will make a purchase, vote for a candidate, etc.
- It relies on the logistic function:

$$P(Y = 1) = \exp(\sum \beta_i X_i) / [1 + \exp(\sum \beta_i X_i)]$$

Logit vs. Probit



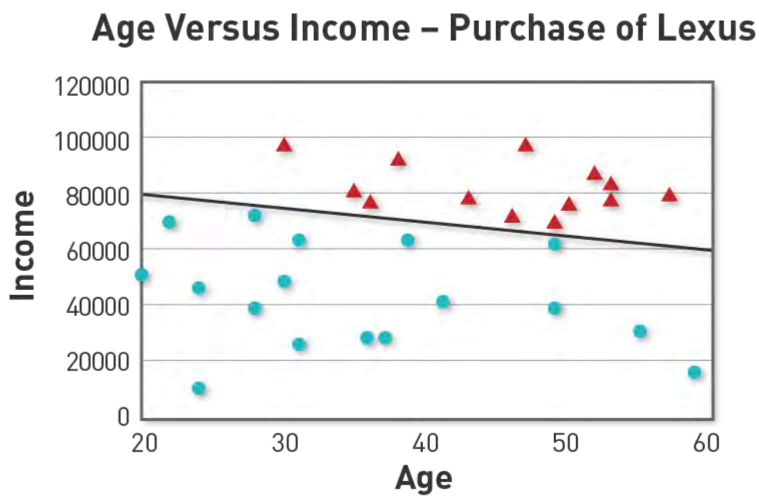
Logit	Probit
• Relies on the logistic distribution. • Is more sensitive to extreme values.	• Relies on the normal distribution. • Is more sensitive to values near the mean.

- Logit and probit usually produce similar results.

View All View Keyframes

Perceptrons

- Perceptrons are an advanced decision-making tool, first developed in the 1960s.
- They focus on choices, where a straight line drawn through data can separate patterns.



- Perceptrons work well for some problems but not for others.

8 of 35

Perceptron Example 1: 'And' Condition

- Consider two Lexus buyer variables: rich (or not rich) and young (or not young).

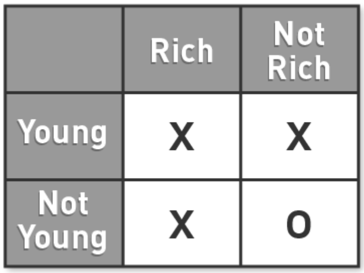
	Rich	Not Rich
Young	X	O
Not Young	O	O

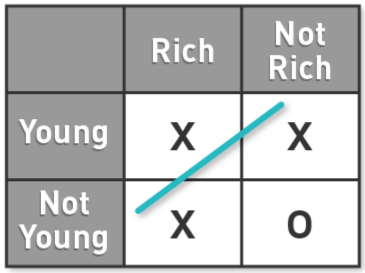
	Rich	Not Rich
Young	X	O
Not Young	O	O

- This example uses an "And" condition: a buyer must be both rich and young.
- With the "And" condition, a straight line can divide the buyers from the nonbuyers.

18 of 35

Perceptron Example 2: 'Or' Condition

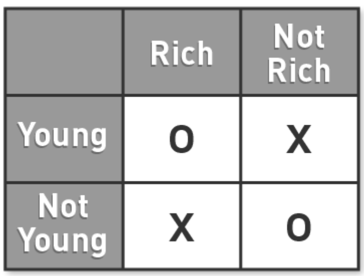


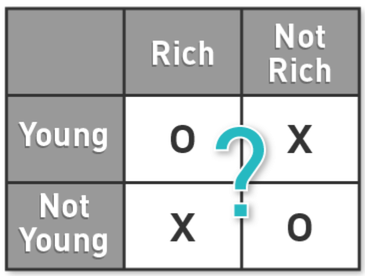


- This example uses an "Or" condition: a buyer must be either rich or young.
- With the "Or" condition, a straight line can divide the buyers from the nonbuyers.

24 of 35

Perceptron Example 3:
'Exclusive Or' Condition





- This example uses an "Exclusive Or" condition: a buyer must be either rich or young, but not both rich and young.
- With the "Exclusive Or" condition, a straight line cannot divide the buyers from the nonbuyers.

30 of 35

Perceptrons to Neural Networks

- Although innovative, perceptrons could not solve the "Exclusive Or" problem (A or B but not both).
- It's failure was reliance on the straight line or linear function.
- Breaking the linearity constraint (i.e., allowing curved lines) solved the problem.
- Applying the logistic function curve led to the development of neural networks.

35 of 35

View All View Keyframes

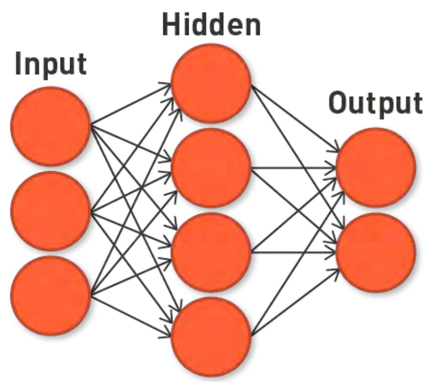
Neural Networks

- Neural networks are a powerful technique to identify trends and patterns; they expand on the concept of perceptrons.
- Neural networks work in a way that is similar to the human brain.

3 of 53

Neural Network Characteristics

- Neural networks use the logistic function to represent nonlinear behavior.
- A "hidden" layer of variables allows modeling of any mathematical representation.
- Neural networks are slow to learn but very effective in decision making.



9 of 53

Neural Network Variables

- Inputs:
  - These are the X variables that would be included in a regression.
- Hidden nodes:
  - These are calculated by the neural network.
  - The user specifies how many hidden nodes to use.
  - More hidden nodes increase the ability of the neural network to find a pattern.
  - More hidden nodes also increase the time required to find a solution.

17 of 53

Neural Network Variables (cont.)

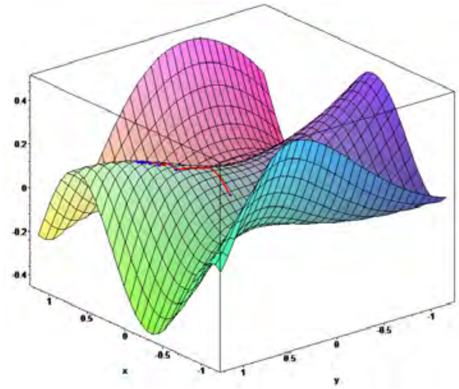
- Outputs:
  - These are the Y variables.
  - There can be more than one Y variable (e.g., "buy a TV" or "buy a cell phone").

20 of 53

Neural Networks: Hill Climbing

- The neural network process is like

climbing a hill.
- From a given starting point: find the direction of maximum slope, take a step in that direction, then repeat.



25 of 53

Neural Networks: Gradient Search

- Gradient search is the mathematical process of hill climbing.
- First, the neural network examines where it is in the landscape.
- Second, it calculates the slope in all directions, by taking the derivative of the logistic function, $f(x)$:

$$f'(x) = f(x) * (1 - f(x))$$

- Third, it takes a step uphill, then repeats the process until it reaches the top of the hill.

32 of 53

Problems With Gradient Search

- Local optima challenge:
  - The problem with gradient search, or hill climbing, is that you might be climbing a small hill, not the tallest hill in the area.
  - That means the process may find a solution that is suboptimal.
- Solution:
  - Run the neural network multiple times with different starting points in the landscape.
  - The neural network package in R randomly selects a new starting point each time.

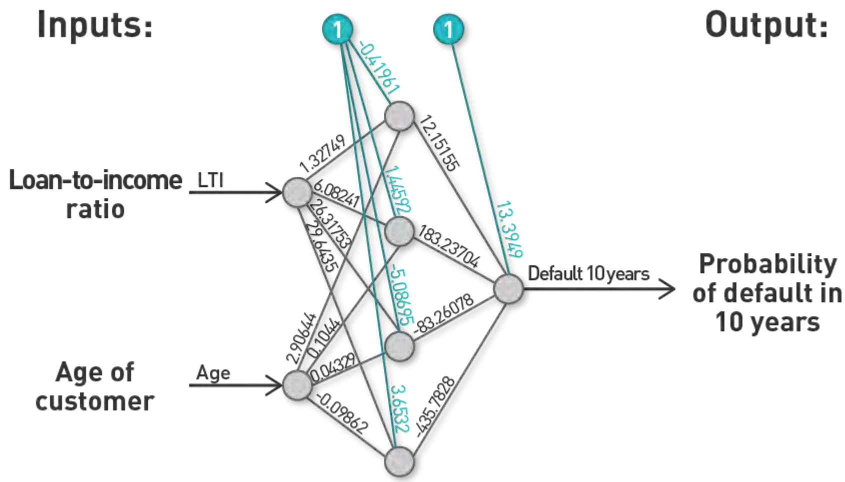
39 of 53

Neural Network Implementation

Step 1:	Collect data.
Step 2:	Identify input variables (i.e., the X variables).
Step 3:	Identify output variables (i.e., the Y variables).
Step 4:	Determine number of hidden nodes.
Step 5:	Run neural network.

45 of 53

Neural Network Solution Example

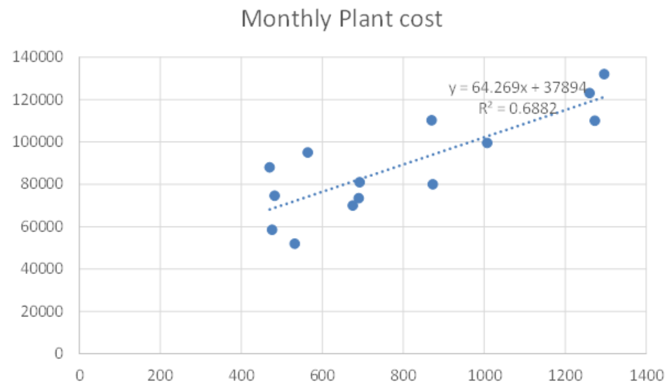

Inputs:
Output:
Loan-to-income ratio
LTI
Age of customer
Age
Default 10years
Probability of default in 10 years
-0.41961
13.3949
1.32749
6.08241
26.31753
29.6435
12.15155
1.44592
183.23704
-5.08695
-83.26078
2.90644
0.1044
0.04329
-0.09862
3.6532
-435.7828

53 of 53

View All View Keyframes

Regression: A Review

Our earlier regression estimated plant cost based on units:

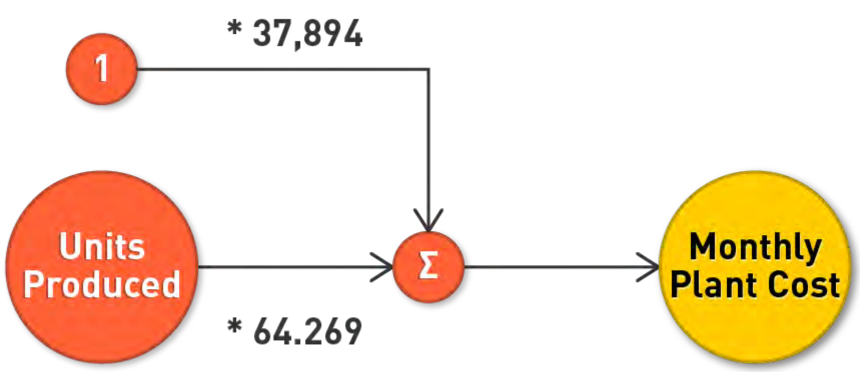


$$Monthly\ Plant\ Cost = 37{,}894 + 64.269 * Units\ Produced$$

4 of 33

Another Way to View Regression: Example 1

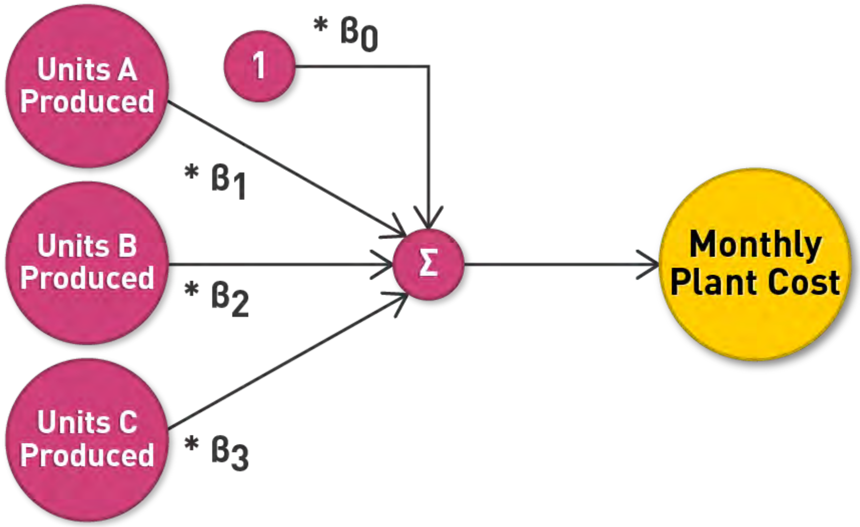
$$Monthly\ Plant\ Cost = 37{,}894 + 64.269 * Units\ Produced$$



11 of 33

Another Way to View Regression: Example 2

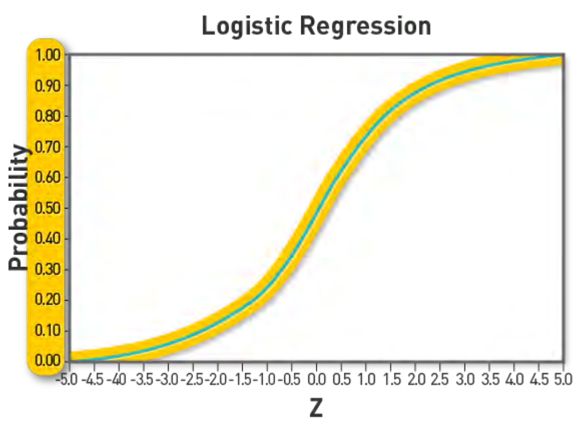
$$Monthly\ Plant\ Cost = \beta_0 + \beta_1 * A\ produced + \beta_2 * B\ produced + \beta_3 * C\ produced$$



18 of 33

Logit: A Different Perspective

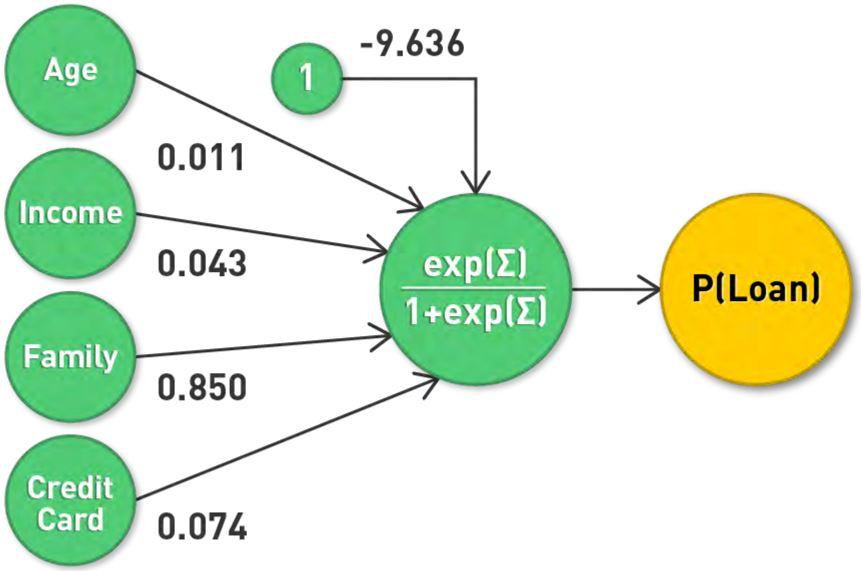
- Let's predict a loan approval based on:
  - Age
  - Income
  - Family size
  - Credit card average balance
- This is a choice model, so we'll use logit.



27 of 33

Another Way to View Logit

$$P(Y=1) = \frac{\exp(-9.636 + 0.011 * Age + 0.043 * Income + 0.850 * Family + 0.074 * CCAvg)}{[1 + \exp(-9.636 + 0.011 * Age + 0.043 * Income + 0.850 * Family + 0.074 * CCAvg)]}$$


Age
1
-9.636
0.011
Income
0.043
Family
0.850
Credit Card
0.074
exp(Σ) / 1+exp(Σ)
P(Loan)

33 of 33