L5 Logistic Regression

Regression Vs. Classification Problems

- Variables can be characterized as either quantitative or qualitative (also known as categorical).
- Quantitative variables take on numerical values.
- Ex: person's age, height, or income, the value of a house, and the price of a stock.
- Qualitative variables take on values in one of K different classes, or categories.
- Ex: the brand of product purchased (brand A, B, or C), whether a person defaults on a debt (yes or no), or a disease diagnosis.
- We tend to refer to problems with a quantitative response as regression problems, while those involving a qualitative response are often referred to as classification problems

Logistic Regression

- Logistic regression is another technique borrowed by machine learning from the field of statistics.
- Logistic regression is named for the function used at the core of the method, the logistic function.
- The logistic function, also called the sigmoid function
- It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

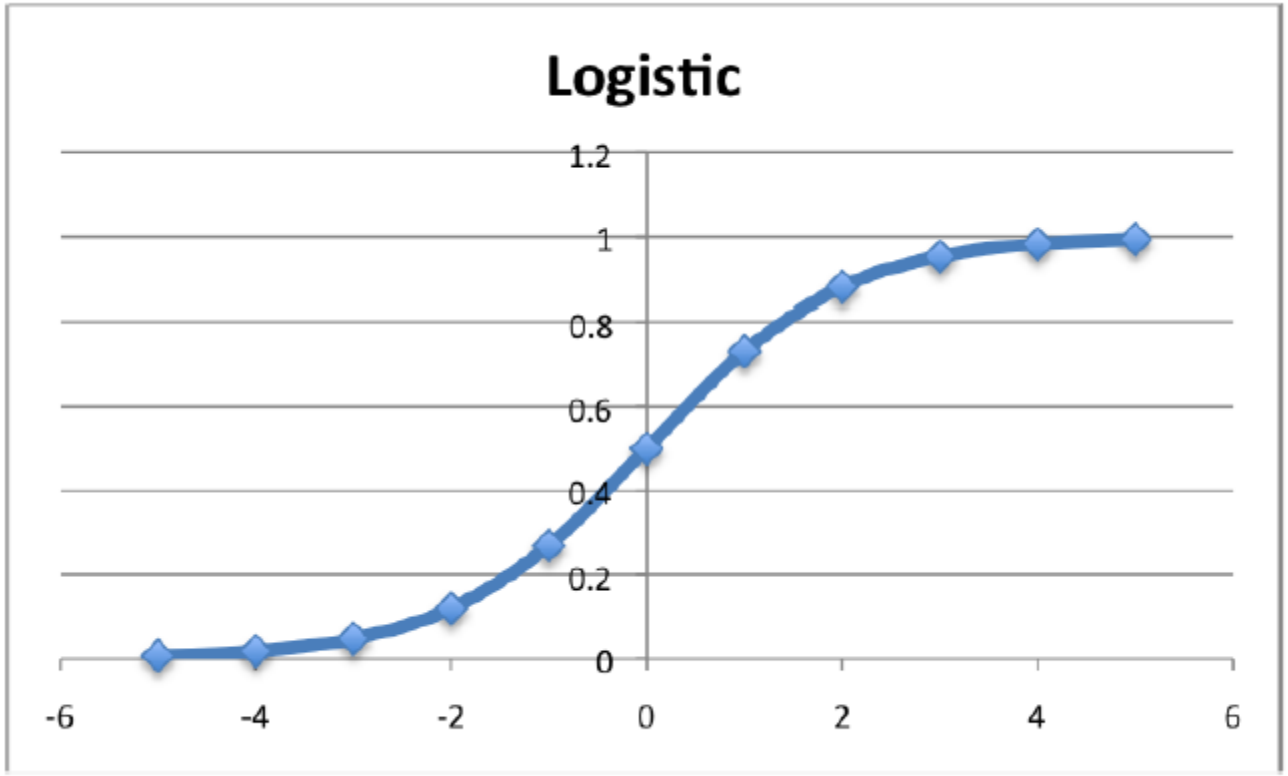
Logistic Regression

- The first thing to note is that logistic regression is not a regression, but a classification learning algorithm
- The name comes from statistics and is due to the fact that the mathematical formulation of logistic regression is similar to that of linear regression
- standard logistic function (also known as the sigmoid function) is:

$$\frac{1}{1+e^{-value}} \qquad f(x) = \frac{1}{1+e^{-x}},$$

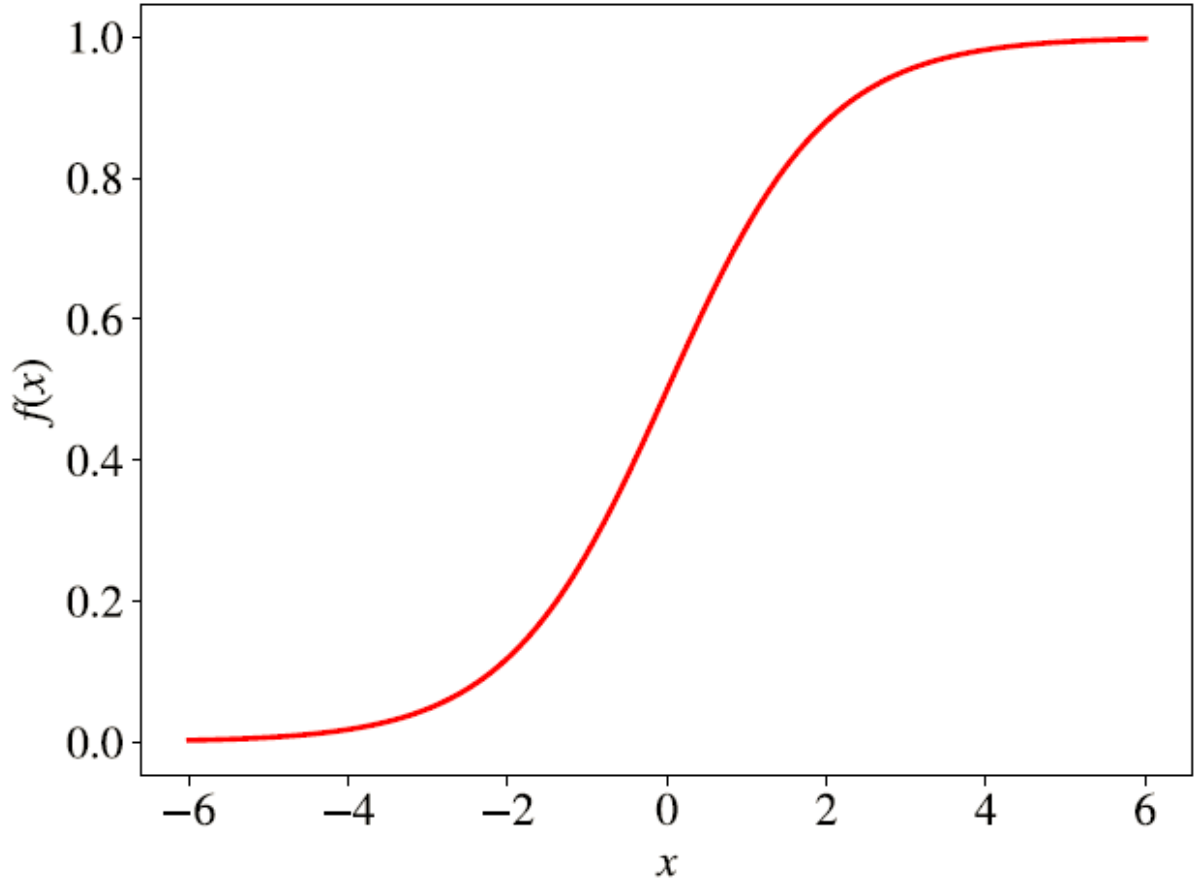
- e is the base of the natural logarithm also called Euler's number;
- e^x is also known as the exp(x) function in programming languages
- ***value*** is the actual numerical value that we want to transform.
- Note: The natural logarithm (ln) and the exponential function e^x are inverse operations
- Plot of the numbers between -5 and 5 transformed into the range 0 and 1 using the logistic function.

Standard logistic function - Graph



- By looking at the graph of the standard logistic function, we can see how well it fits our classification purpose:
- we could interpret the output of f(x)
  - if it's higher than or equal to the threshold 0.5 we would say that the class of x is 1 or positive;
  - otherwise, it's 0 or negative

Standard logistic function - Graph




```
# x is any real number
import numpy as np
print(1.0 / (1 + np.exp(-x)))
```

Define function:

```
def sigmoid(x):
        return 1.0 / (1 + np.exp(-x))
```

Ex: exp(0)= 1 and sigmoid(0)= 0.5

Sigmoid Function

- The logistic function in linear regression is a type of sigmoid
- Sigmoid is a mathematical function that takes any real number and maps it to a probability between 1 and 0.
- The sigmoid function forms an S shaped graph
  - that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.
  - which means as x approaches infinity, the probability becomes 1, and as x approaches negative infinity, the probability becomes 0
- The formula of the sigmoid function is:

$$f(x) = \frac{1}{1 + e^{-x}},$$

Logistic Regression

- Logistic regression is one of the most popular machine learning algorithms for binary classification.

$$f(x) = \frac{1}{1 + e^{-x}},$$

  - f(x) = y = output between 0 and 1 (probability estimate)
  - x = input to the function (our algorithm's prediction ex: B0+ B1 * x).
  - Denote x as z to distinguish such that $y= 1/(1+e^{-z})$ where $z= B0+B1*x$
  - e = base of natural log (ln is denoted for base e. Ex: log of base e is $\log_e$ = ln (natural log))
- if the value of x goes to positive infinity then the predicted value of y will become 1
- if it goes to negative infinity then the predicted value of y will become 0.
- And if the outcome of the sigmoid function is more than 0.5 then we classify that label as class 1 or positive class
- if it is less than 0.5 then we can classify it to negative class or label as class 0

How Logistic Regression Predicts Probabilities

- Logistic regression models the probability of the default class (e.g. the first class).
- Ex: if we are modeling people's status (Y) as rich(1) or poor(0) from their income(X), then the first class could be rich
- The logistic regression model could be written as the probability of rich given a person's income
- Formally: P(income) = P(status = rich|income)
- Another way, we are modeling the probability that an input (X) belongs to the default class (Y = 1), we can write this formally as:
- P(X) = P(Y = 1|X)
- P(y=1|X) is the probability that the output y is 1 given input X

Derivation of sigmoid function

- 1. Sigmoid function is defined as P(Y = 1|X) = 1/(1+e^(-z))
  - Sigmoid function is related to the probability
  - The output of the sigmoid function can be directly interpreted as the estimated probability P(y=1|x).
  - Ex: if the sigmoid function outputs 0.8 for a given input, it means that the algorithm estimates an 80% chance that the event y=1 (belonging to a certain class) occurs given the input x
- 2. Derive 1/(1+e^(-z)) = e^z/(1+e^z))

- Note: z is a linear combination of input features and weights used in logistic regression. z= B0+B1*x

Logistic Regression

- Logistic regression uses an equation very much like linear regression.
- Input values (x) are combined linearly using weights or coefficient values to predict an output value (y).
- A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.
- Below is an example logistic regression equation:

$$y = \frac{e^{B0+B1\times x}}{1+e^{B0+B1\times x}}$$

  - y is the predicted output,
  - B0 is the bias or intercept term and
  - B1 is the coefficient for the single input value (x).
- Each column in our input data has an associated B coefficient (a constant real value) that must be learned from our training data.

How Logistic Regression Predicts Probabilities

- logistic regression is a classification algorithm
- Note that the probability prediction must be transformed into a binary values (0 or 1) to actually make a prediction.
- Logistic regression is a linear method, but the predictions are transformed using the logistic function.
- The impact of this is that we can no longer understand the predictions as a linear combination of the inputs as we can with linear regression
- Continuing on from above, the model can be stated in terms pf p(x) rather than y as termed previously

$$y = \frac{e^{B0+B1\times x}}{1+e^{B0+B1\times x}} \qquad p(X) = \frac{e^{B0+B1\times X}}{1+e^{B0+B1\times X}}$$

- Note: Sigmoid function is related to the probability
- The output of the sigmoid function can be directly interpreted as the estimated probability $P(y=1|x)$.
- Ex: if the sigmoid function outputs 0.8 for a given input, it means that the algorithm estimates an 80% chance that the event y=1 (belonging to a certain class) occurs given the input *x*

Odds and log odds

- In probability, odds represent the ratio of the probability of success (an event occurring) to the probability of failure (the event not occurring).
- If P is the probability of success and Q is the probability of failure (Q=1−P), then the odds O =P/Q= P/(1-P)
- The value of Odds range from zero to infinity and the value of probability lie between zero and one.
- Ex: For example, if the probability of winning a game is 0.6, then the odds of winning = 0.6/0.4=0.6/0.4=1.5.
- Log odds, often referred to as logit, is the natural logarithm of the odds. Mathematically, it is represented as
- $\text{Logit}(O)=\ln(P/Q)$

Odds and log odds - Background

- Why Odds ratio P/Q is transformed into ln(P/Q)
- It transforms the odds ratio into a linear scale
- making it more amenable to modeling using linear regression techniques.
- The log odds have the property that they range from negative infinity to positive infinity.
- The log odds also provide a way to measure the strength of the relationship between predictor variables and the outcome.
- Changes in the log odds can be more interpretable than changes in probabilities, especially when dealing with binary outcomes.

Odds and log odds - Summary

- The expression ln(P(x)/(1−$P(x)$) is the natural logarithm of the odds ratio
- $P(x)$: This is the predicted probability that a specific event (often denoted as y=1) occurs given the input x. It's the output of the sigmoid function and represents the estimated likelihood of the event happening.
- 1−$P(x)$: This is the complementary probability that the event does not occur. Since probabilities add up to 1, 1−$P(x)$ represents the estimated likelihood of the event not happening.
- (P(x)/(1−$P(x)$: This ratio is the odds of the event happening (success) divided by the odds of the event not happening (failure). It's a measure of the relative likelihood of the event occurring versus not occurring
- ln: This represents the natural logarithm function. Taking the natural logarithm of the odds ratio is a common transformation used in logistic regression to convert the odds ratio into the log-odds, which is a linear measure

Simplify log odds expression

- Simplifying ln(P(x)/(1−*P*(*x*))
- We have sigmoid function given by P(x) = e^z/(1+e^z)
- 1-p(x)= 1/(1+e^z)
- ln(P(x)/(1−P(x)) =ln(e^z) = z
- ln(P(x)/(1−*P*(*x*)) = z
- Note that Input values (z) are combined linearly using weights or coefficient values to predict an output value (y).
- z = B0+B1*x
- ln(P(x)/(1−P(x)) =B0+B1*x
- ln(odds) = B0 + B1 * X

How Logistic Regression Predicts Probabilities

- Simplifying

$$ln(\frac{p(X)}{1-p(X)}) = B0 + B1 \times X$$

- the calculation of the output on the right is linear again just like linear regression
- the input on the left is a natural logarithm of the probability of the default class (y=1).
- This ratio on the left is called the odds of the default class
- Odds are calculated as a ratio of the probability of the event divided by the probability of not the event,
- Ex: 0.8/(1-0.8) as the odds of 4
- ln(odds) = B0 + B1 * X
- It is common to refer to the transform that relates the linear regression equation to the probabilities as the link function, e.g. the logit link function.

How Logistic Regression Predicts Probabilities

- Simplifying the expression ln(odds)=B0+B1*X
- Exponentiate both sides of the equation.
- Note: natural logarithm (ln) and the exponential function (e^x) are inverse operations
- odds = $e^{B0+B1*X}$
- Summarizing:
- So, the model is a linear combination of the inputs
- but this linear combination relates to the log-odds of the default class.
- The transformation from the linear combination z=B0+B1*x to the odds is the basis for understanding how logistic regression models predict the likelihood of an event happening based on input features.

Learning the Logistic Regression Model

- The coefficients of the logistic regression algorithm must be estimated from our training data.
- This is done using maximum-likelihood estimation.
- The best coefficients would result in a model that would predict
  - a value very close to 1 (ex: status = rich) for the default class and
  - a value very close to 0 (e.g. poor) for the other class.
- The intuition for maximum-likelihood for logistic regression
- is that a search procedure seeks values for the coefficients that minimize the error in the probabilities predicted by the model (ex: probability of 1 if the data is the primary class)
- This means that a minimization algorithm is used to optimize the best values for the coefficients for our training data.
- This is often implemented in practice using efficient numerical optimization algorithm
- we can implement from scratch using the gradient descent algorithm

Making Predictions with Logistic Regression

- Problem statement: Assume we have a model that can predict whether a person is rich or poor based on their income
- Given an income height of 150 hundreds is the person rich or poor?
- Assume that we have learned the coefficients of B0 = -100 and B1 = 0.6.
- We can calculate the probability of rich given an income of 150 hundreds
- Formally P(rich | income = 150)
- Calculate y
- B0=-100, B1= 0.6, x= 150

Making Predictions with Logistic Regression

- Problem statement: Assume we have a model that can predict whether a person is rich or poor based on their income
- Given an income height of 150 hundreds is the person rich or poor?
- Assume that we have learned the coefficients of B0 = -100 and B1 = 0.6.
- Formally P(rich | income = 150)

$$y = \frac{e^{B0+B1\times X}}{1+e^{B0+B1\times X}}$$

- Y= 0.000453978687.
- So a probability of near zero that the person is rich
- prediction = 0 IF p(rich) < 0.5
- prediction = 1 IF p(rich) >= 0.5

Prepare Data for Logistic Regression

- The assumptions made by logistic regression about the distribution of data is much the same as the assumptions made in linear regression.
- in predictive modeling we are focused on making accurate predictions
- Binary Output Variable: logistic regression is intended for binary (two-class) classification problems.
- It will predict the probability of an instance belonging to the default class, which can be snapped into a 0 or 1 classification.

Prepare Data for Logistic Regression

- Remove Noise: Logistic regression assumes no error in the output variable (y), consider removing outliers and possibly misclassified instances from training data.
- Gaussian Distribution: Logistic regression is a linear algorithm (with a nonlinear transform on output).
- It does assume a linear relationship between the input variables with the output.
- Data transforms of our input variables that better expose this linear relationship can result in a more accurate model.

Prepare Data for Logistic Regression

- Remove Correlated Inputs: Like linear regression, the model can overfit if we have multiple highly-correlated inputs.
- Consider calculating the pairwise correlations between all inputs and removing highly correlated inputs.
- Fail to Converge: It is possible for the expected likelihood estimation process that learns the coefficients to fail to converge.
- This can happen if there are many highly correlated inputs in our data or the data is very sparse (e.g. lots of zeros in input data).

Logistic Regression by Stochastic Gradient Descent with two input variables

Training data

x1	x2	y
2	1	1
3	2	1
4	3	0
5	4	0

Logistic Regression by Stochastic Gradient Descent

1. Initialize Coefficients: B0= 0, B1=0, B2=0, alpha=0.01
2. Perform SGD
2.1 Calculate z = B0+B1*x1+B2*x2
2.2 Apply the sigmoid function to get the prediction P(y=1|x)= 1/(1+e^(-z))
2.3 Calculate error= y-prediction= y-P(y=1|x)
2.4 update the coefficients using SGD Update Equation
b = b + alpha * (y - prediction) * prediction * (1 - prediction)* x
Update Equation for B0, B1 and B2
B0 = B0 + alpha * (y - prediction) * prediction * (1 - prediction)* 1
B1 = B1 + alpha * (y - prediction) * prediction * (1 - prediction)* x1
B2 = B2 + alpha * (y - prediction) * prediction * (1 - prediction)* x 2

Logistic Regression by Stochastic Gradient Descent

- **Row 1:**
- X1=2, x2=1, y=1
- Z= 0 and P(y=1|x)=0.5
- Error = 1-0.5=0.5
- Update coefficients
- B0=0.0025
- B1= 0.005
- B2= 0.0025
- Repeat for every row. One complete iteration of training data forms one epoch.

Logistic Regression by Stochastic Gradient Descent

- Get the final value of B0, B1 and B2 after one (specified number of epochs) epoch
- Now that we have trained the model, we can use it to make predictions.
- Using coefficients, get the updated prediction by applying sigmoid function
- Get the new target label

Logistic Regression by Stochastic Gradient Descent

- Repeat the Process
- repeat this process and update the model for each training instance in the dataset.
- A single iteration through the training dataset is called an epoch.
- It is common to repeat the stochastic gradient descent procedure for a fixed number of epochs.
- At the end of epoch, calculate error values for the model.
- Because this is a classification problem, we need to get an idea of how accurate the model is at each iteration.
- Plot the graph of accuracy of the model over specified epochs

  Plot epoch (x axis) vs accuracy (y axis)

Logistic Regression by Stochastic Gradient Descent

- Compute accuracy
- prediction = IF (output < 0.5) Then 0 Else 1
- With this simple procedure we can convert all of the outputs to class values (or target labels):
- Finally, we can calculate the accuracy for the model on the training dataset:
- Accuracy = (Correct predictions/Total predictions) * 100
- Total predictions = number of training instances = 4
- For the training data, calculate accuracy

Problem Statement 2

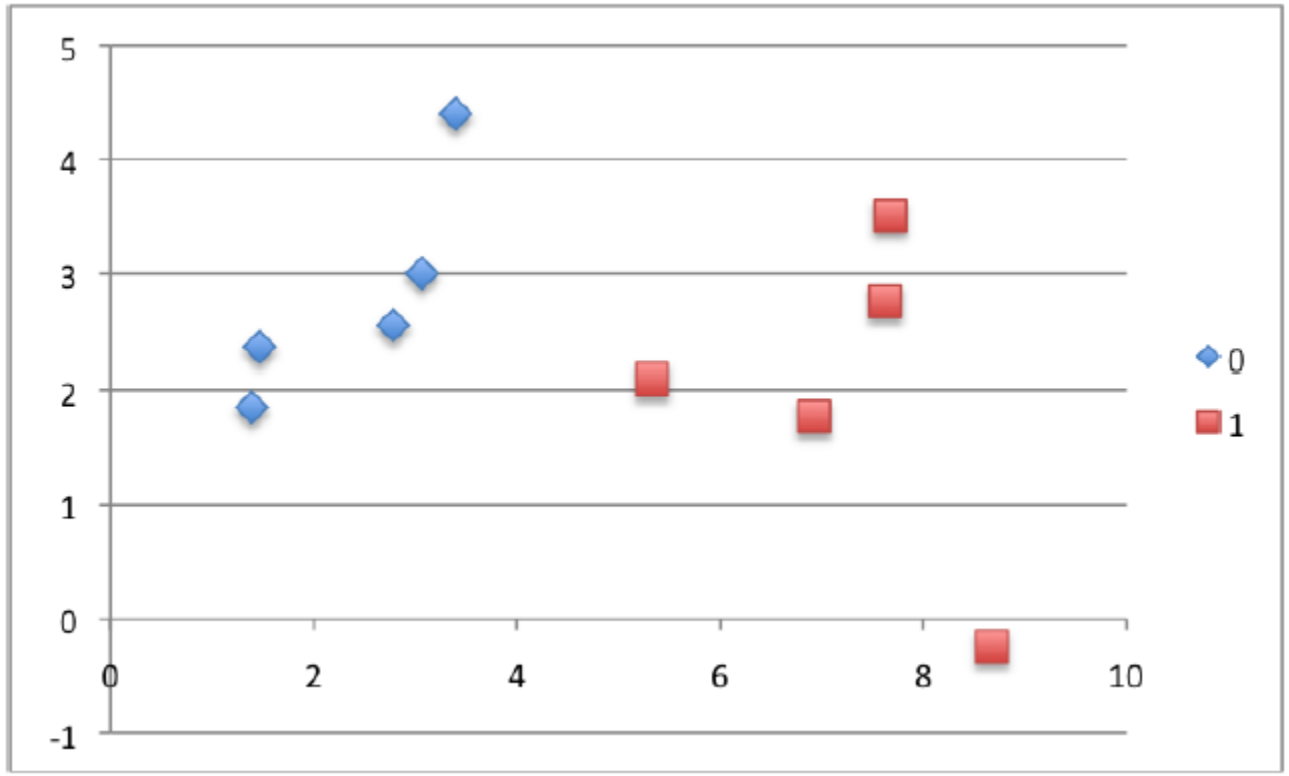
Training dataset

- This dataset has two input variables (X1 and X2) one output variable (Y).
- The input variables are real-valued random numbers drawn from a Gaussian distribution
- The output variable has two values, making the problem a binary classification problem
- Plot the graph of X1 (x- axis) vs X2 (y axis)

X1	X2	Y
2.7810836	2.550537003	0
1.465489372	2.362125076	0
3.396561688	4.400293529	0
1.38807019	1.850220317	0
3.06407232	3.005305973	0
7.627531214	2.759262235	1
5.332441248	2.088626775	1
6.922596716	1.77106367	1
8.675418651	-0.242068655	1
7.673756466	3.508563011	1

Logistic Regression Dataset – X1 vs X2

How do you analyse the graph?



Logistic Regression

- We can easily draw a line to separate the classes
- For this dataset, the logistic regression has three coefficients just like linear regression
- output = B0 + B1 * X1 + B2 * X2
- The job of the learning algorithm will be to discover the best values for the coefficients (B0, B1 and B2) based on the training data.
- Unlike linear regression, the output is transformed into a probability using the logistic function:

$$p(class = 0) = \frac{1}{1 + e^{-output}}$$

Logistic Regression by Stochastic Gradient Descent

- We can estimate the values of the coefficients for the logistic regression model using stochastic gradient descent
- Calculate Prediction
- Assign 0.0 to each coefficient and calculating the probability of the first training instance that belongs to class 0.
- B0 = 0.0
- B1 = 0.0
- B2 = 0.0
- The first training instance is: X1 = 2.7810836, X2 = 2.550537003, Y = 0.
- Using the above equation, calculate a prediction: $$prediction = \frac{1}{1 + e^{-(B0 + B1 \times X1 + B2 \times X2)}}$$

Logistic Regression by Stochastic Gradient Descent

- calculate a prediction

$$prediction = \frac{1}{1 + e^{-(B0+B1\times X1+B2\times X2)}}$$

$$prediction = \frac{1}{1 + e^{-(0.0+0.0\times 2.7810836+0.0\times 2.550537003)}}$$

$$prediction = 0.5$$

Logistic Regression by Stochastic Gradient Descent

- Calculate New Coefficients
- We can calculate the new coefficient values using update equation.
- b = b + alpha * (y - prediction) * prediction * (1 - prediction)* x
- b - the coefficient we are updating
- Prediction - the output of making a prediction using the model.
- Alpha - learning rate and controls how much the coefficients (and therefore the model) changes or learns each time it is updated.
- Note: Good values might be in the range 0.1 to 0.3. Here, use a value of 0.3.

Logistic Regression by Stochastic Gradient Descent

- Update the coefficients using the prediction (0.5) and coefficient values (0.0), alpha = 0.3
- Update Equation
- b = b + alpha * (y - prediction) * prediction * (1 - prediction)* x
- Update Equation for B0, B1 and B2
- B0 = B0 + alpha * (y - prediction) * prediction * (1 - prediction)* 1
- B1 = B1 + alpha * (y - prediction) * prediction * (1 - prediction)* x1
- B2 = B2 + alpha * (y - prediction) * prediction * (1 - prediction)* x2

Note: x is the input value for the coefficient.
B0 does not have an input.
This coefficient is called the bias or the intercept and we can assume B0 always has an input value of 1.0.

Logistic Regression by Stochastic Gradient Descent

$$B0 = 0.0 + 0.3 \times (0 - 0.5) \times 0.5 \times (1 - 0.5) \times 1.0$$
$$B1 = B1 + 0.3 \times (0 - 0.5) \times 0.5 \times (1 - 0.5) \times 2.7810836$$
$$B2 = B2 + 0.3 \times (0 - 0.5) \times 0.5 \times (1 - 0.5) \times 2.550537003$$

$$B0 = -0.0375$$
$$B1 = -0.104290635$$
$$B2 = -0.095645138$$

Applying stochastic gradient descent to the problem of finding the coefficients for the logistic regression model

- Given each training instance:
  - Calculate a prediction using the current values of the coefficients.
  - Calculate new coefficient values based on the error in the prediction.
- The process is repeated until the model is accurate enough (e.g. error drops to some desirable level) or for a fixed number iterations

Logistic Regression by Stochastic Gradient Descent

- Repeat the stochastic gradient descent procedure for 10 epochs.
- At the end of epoch calculate error values for the model.
- Because this is a classification problem, get accuracy of the model at each iteration.
- Plot the graph to show accuracy of the model over 10 epochs.

Logistic Regression with Gradient Descent - Accuracy versus Iteration