Logistic Regression

Sean Kang

What is Logistic Regression

- One of a type of classification algorithm (many other exists)
- Analyzes the relationship between the dependent variable with one or more independent variable. Also used for prediction.
- This measures the coefficients between the dependent variable with the one or more independent variable.
- It is used to measure the probabilities of the classification result.

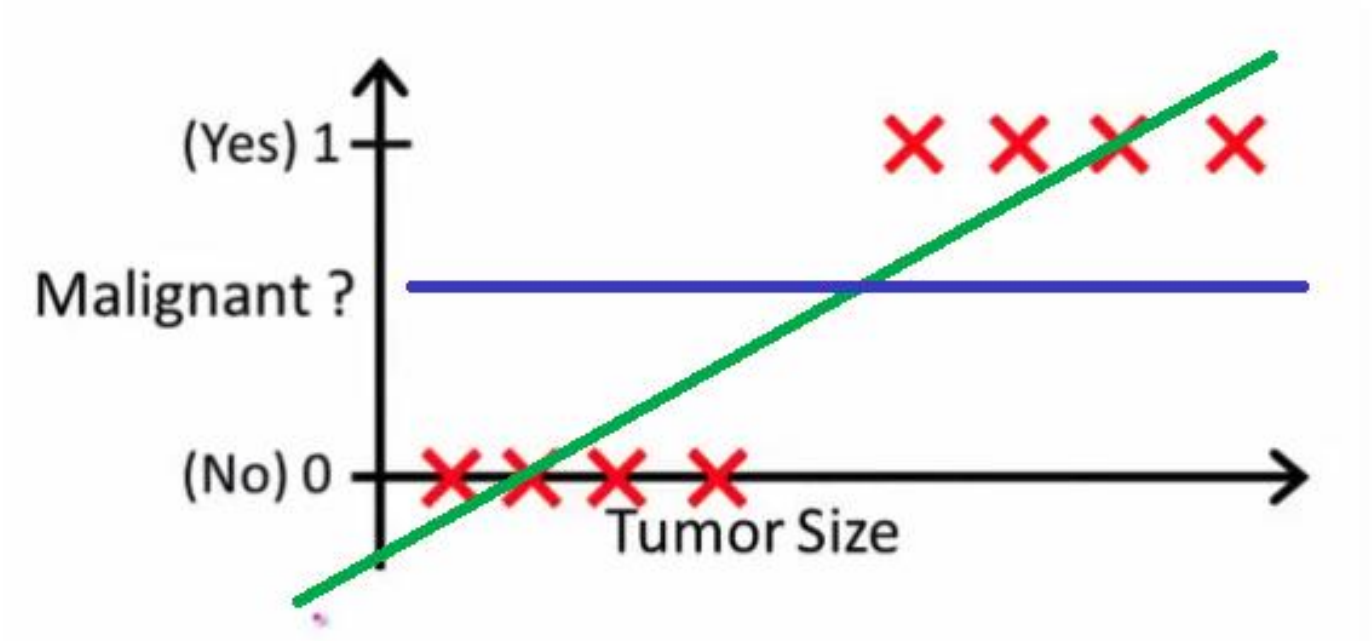
  - Based on the spread and over/under, the implied score for the game is Boilermakers 78, Wolfpack 68.
  - The Boilermakers have an 81.5% chance to collect the win in this matchup per the moneyline's implied probability.
  - The Wolfpack sit with a 22.7% implied probability to come out on top.

Multiple types of Logistic Regression

- Binomial (binary) Logistic Regression – analyzes the relationship with a dependent variable that has a binary outcome. It is a regression analysis when the outcome or dependent variable is binary – dichotomous (true/false, die/live, buyer pays/no-sale, pass/fail…).
- Multinomial Logistic Regression – analyzes the relationship on an outcome or dependent variable which has three or more values (example, the type of product purchased, type of injury, type of animal recognized by image scanner). The outcome does not have a rank.
  - No: low, medium, high
  - Yes: mammal, amphibian, fish, bird, etc
- Ordinal Logistic Regression – predicts the probability that an outcome or dependent variable generates into an ordinal value (severity of the cancer, level of customer satisfaction (low, medium, high), etc)

Why use Logistic Regression over Linear Regression?

- In linear regression, the outcome or output is some scalar value such as MPG.
- But logical regression, the output is a value such as True/False, Buy/Pass, Win/Lose, etc.



How does Logistic Regression work?

- Similar General Steps
- 1.) Prepare the data
- 2.) Train the model
- 3.) Evaluate the model
- 4.) Use the model on predictions

The Math behind Logistic Regression

- In a binary classification case the formula for the regression function is:

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

- Z is a linear function which looks like this:

$z = b_0 + b_1x_1 + \ldots + b_rx_r$

Where:

- b_0, b_1 ...b_r are the model's **predicted weights** or **coefficients**.
- **x** the feature values.

More Math

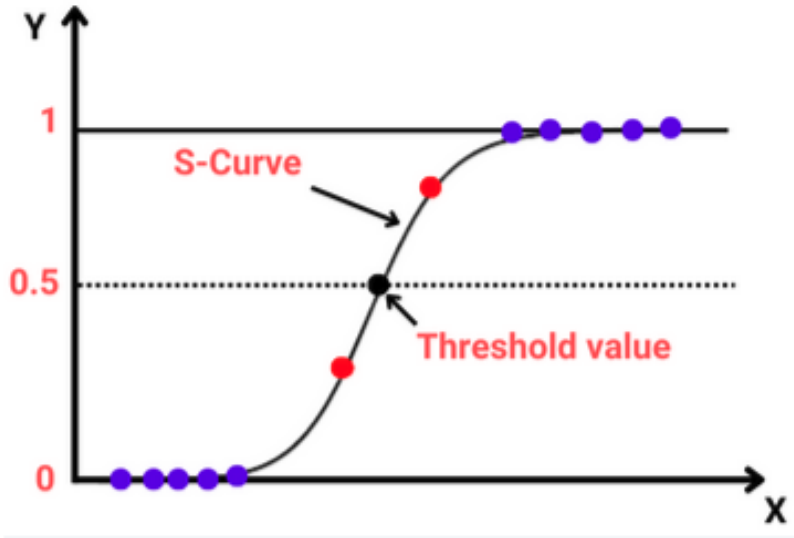
- The z is also known as the log loss function

$$z = log\left(\frac{p(x)}{1 - p(x)}\right)$$

- p(x) is the probability of success, 1-p(x) is the prob of failure

Aspects of the Logistic Regression

- Binomial – the outcome is a value of 1 or 0.
- However, the model will generate a probability value between 1 and 0, for Binomial situations.
- Unlike linear regression model, the logistic regression uses a S-shape logistical function –Sigmoid function.
- Threshold value is used

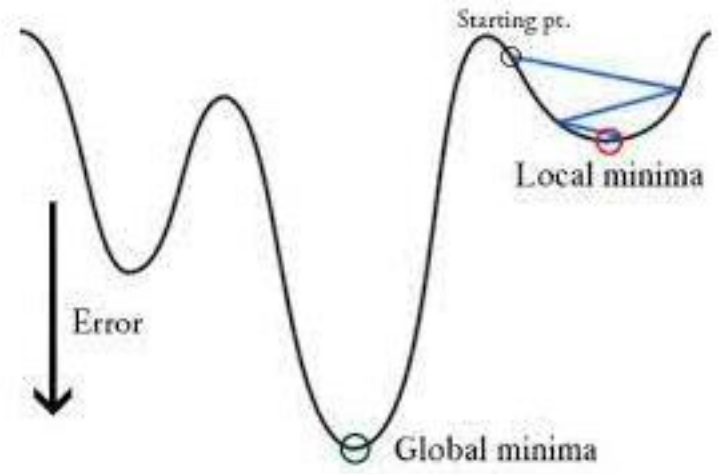


Assumptions

- Responses are binary or multinomial
- Independent variables have no collinearity
- No extreme outliers
- Linear relationship between input variables and log-odds of the response
- Large sample size

How to measure fitness

- With a linear regression, MSE or RMSE is used.
- If MSE is used for a logical regression, local minima is found, and the global minima is missed.



Loss function for Logistic Regression

- Log loss is used for measuring the error in the model, derived from the maximum likelihood estimator (MLE)

$$\text{Log loss} = \frac{1}{N} \sum_{i=1}^{N} -(y_i * \log (\hat{Y}_i) + (1- y_i) * \log (1-\hat{Y}_i))$$

- This involves finding the best coefficients to fit the sigmoid curve.

How to do this with SciKit Learn

- Demo
- SciKit Learn has a LogisticRegression library
- We should know that there are hyperparameters that are important
  - Solvers
  - C
  - Penalty

Hyperparameters

- Solver -the algorithm used for optimize the problem
  - Default used in SciKit is lbfgs
  - Lbfgs has decent performance and works well in most cases.
  - Sag works best for large datasets
  - Saga works better for multinomial
  - Liblinear works when you have a many features and dimensions.
- Penalty - works to avoid overfitting, default is l2.
  - The various choices work with specific solvers (l1, l2, elasticnet, etc)
- C – regularization strength – default is 1.
  - Positive float value
  - Low value -> stronger regularization, Higher value -> give more priority on training data