Building regression models

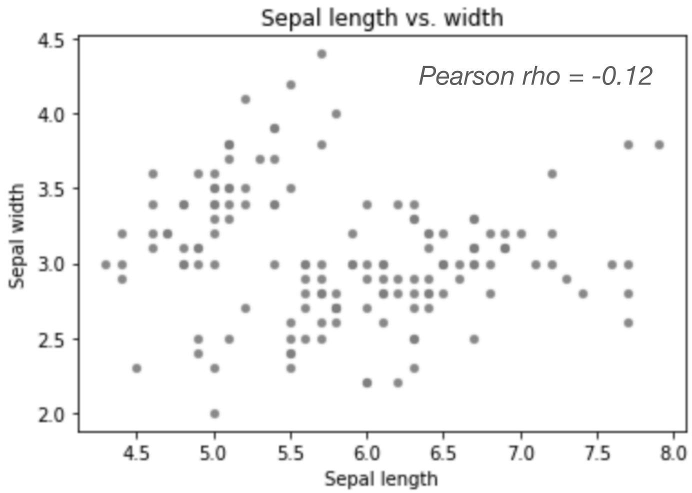
Yulia Newton, Ph.D.
CS156, Introduction to Artificial Intelligence
San Jose State University
Spring 2021

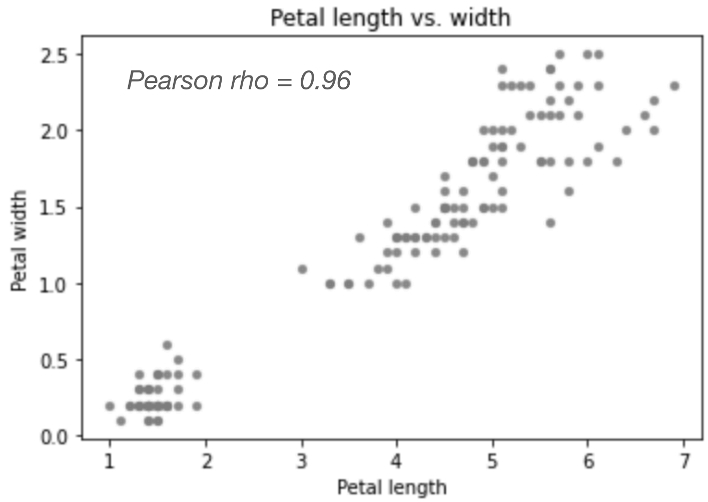
Brief history of regression

- The term first coined by Sir Francis Galton in his "Regression Towards Mediocrity in Hereditary Stature" 1886 publication in *The Journal of the Anthropological Institute of Great Britain*
  - Became known as regression to the mean
  - Pretty cool history of regression description here: http://people.duke.edu/~rnau/regintro.htm

Linear relationship between two variables

- We can always ask "what is the relationship between these two numeric variables?"





Differrent variables correlate differently to each other and the output variable

In Iris dataset the output variable is a 3-class categorical variable

Utilize one or more numeric variables to predict an output numeric variable

- For example:
  - Predict crop yield based on the amount of water and fertilizer used
  - Predict weight loss based on the total calories consumed and the number of hours of aerobic exercise a week
  - Predict blood pressure in patients based on the medication dosage
  - Predict increase in the revenue based on the amount spent on advertising

Regression prediction problem

$$revenue = \beta_0 + \beta(ad\ spending)$$

$$blood\ pressure = \beta_0 + \beta(dose)$$

$$yeild = \beta_0 + \beta_1(water) + \beta_2(fertilizer)$$

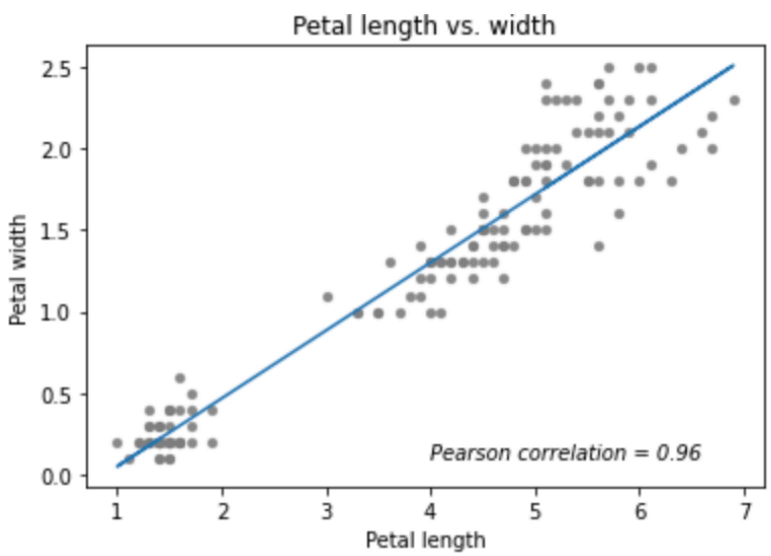
$$weight\,loss = \beta_0 + \beta_1(cal) + \beta_2(exercise)$$

Regression to the rescue

- Regression helps us find general patterns in the data
  - Techniques for finding and analyzing trends in dependent variable vs. independent variables
  - As with other ML problems, the assumption is that the data sample is representative of the general population
    - E.g. sampling only among the basketball players does not give us the representative distribution of height within the general population
- Regression line = best fit line to the training data
  - Best represents general trends in the overall training data

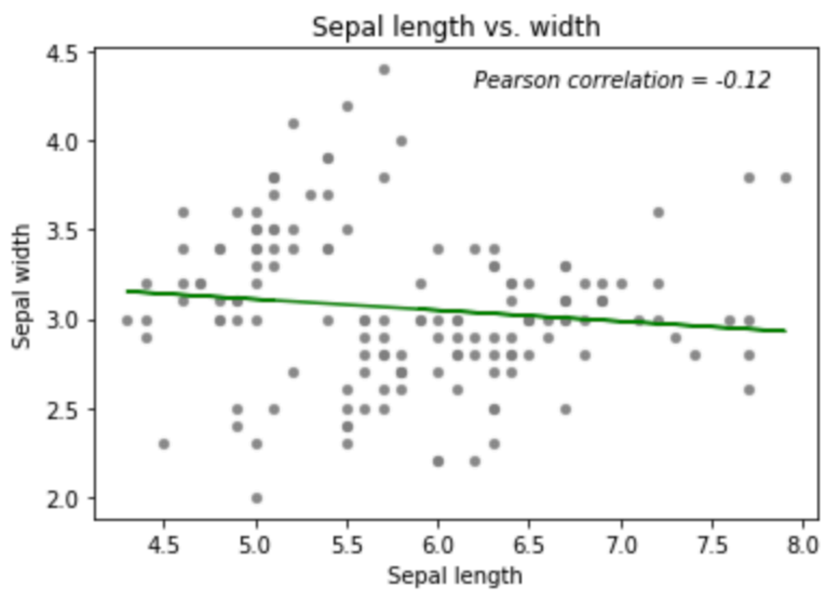
Linear regression is simply a linear function that best fits the training data

- A linear function can be defined by it's slope (m) and its intercept (b)
  - If the slope is positive then there's a positive relationship between the independent and dependent variables (positively correlated)
  - If the slope is negative then the relationship is negative (negatively correlated)
  - Intercept is the expected value of the dependent variable when the value of the independent variable is zero



$$Y = mX + b$$

What happens if there is no relationship between the variables?



Overview of a linear regression model

- Linear regression is based on the idea that contributions of different independent variables to the prediction of a dependent variable are additive

Dataset: $\{y_i,\, x_{i1}, \ldots, x_{ip}\}_{i=1}^{n}$

Unobserved noise (random error)

Independent variables

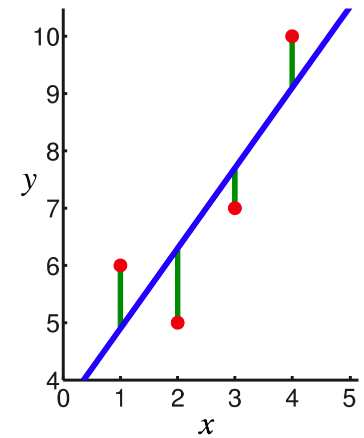
$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\mathsf{T} \boldsymbol\beta + \varepsilon_i, \qquad i = 1, \ldots, n,$$

Dependent variable

Slopes for individual independent variables

Intercept

Matrix notation: $\mathbf{y} = X\boldsymbol\beta + \boldsymbol\varepsilon,$



Find the line that is the best fit across all the predictions in the dataset

https://en.wikipedia.org

Single independent variable example

Prediction for observation i

Line intercept

Line slope

$$Y_i = \beta_0 + \beta_1 X + \epsilon_i$$

Random error for observation i

Dependent variable Y

Independent variable for observation i

Simple regression - single independent variable
Multiple regression - multiple independent variable

Single independent variable example

Output for observation i

Line intercept

Line slope

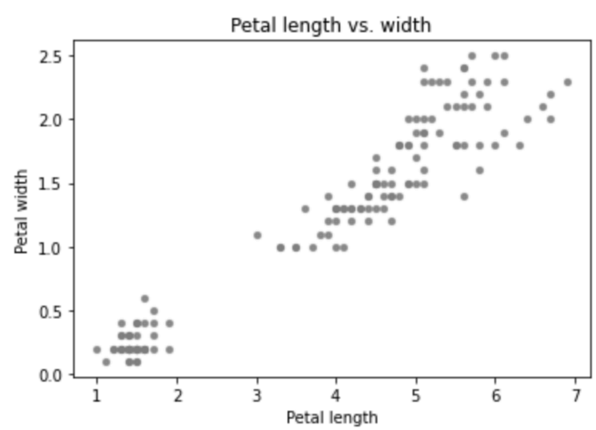
$$Y_i = \beta_0 + \beta_1 X + \epsilon_i$$

Random error for observation i

Dependent variable Y

Independent variable for observation i

Single independent variable example (predict petal width based on petal length):



Single independent variable example

Model coefficients

$$Y_i = \beta_0 + \beta_1 X + \epsilon_i$$

Linear component *Random error component*

Single independent variable example

$$Y_i = \beta_0 + \beta_1 X + \epsilon_i$$

$$\downarrow$$

Applying this model to 1 … i observations

$$\begin{aligned} y_1 &= \beta_0 + \beta_1 x_1 + \epsilon_1 \\ y_2 &= \beta_0 + \beta_1 x_2 + \epsilon_2 \\ &\vdots \\ y_n &= \beta_0 + \beta_1 x_n + \epsilon_n \end{aligned}$$

https://online.stat.psu.edu/stat462/node/132/

Generalizing to multiple independent variables

Slopes for independent variables 1…m

$$Y_i = \beta_0 + \beta_1 X_{1i} + ... + \beta_m X_{mi} + \epsilon_i$$

Generalizing to multiple independent variables

Slopes for independent variables 1…m

$$Y_i = \beta_0 + \beta_1 X_{1i} + \ldots + \beta_m X_{mi} + \epsilon_i$$

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 X_1 \\ \beta_0 + \beta_1 X_2 \\ \vdots \\ \beta_0 + \beta_1 X_n \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Matrix notation

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

https://mnshankar.wordpress.com/2011/05/01/regression-analysis-using-php/

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon$$

https://online.stat.psu.edu/stat462/node/132/

More on random error

- Our regression model does not account for all possible factors affecting the dependent/outcome variable
  - Real life is messy and there are an infinite number of factors that may impact the outcome
- Random noise represents all the factors not included into the regression model

This model does not account for soil type and quality, other vegetation, climate, etc.

$$yeild = \beta_0 + \beta_1(water) + \beta_2(fertilizer)$$

Fitting a linear regression model

- But … that equation is for the global population trend
- These are parameters of the global linear regression model:

$$\beta_0 \ , \ \beta_1 \ \dots \ \beta_m \ , \ \epsilon$$

- These are estimated parameters of our linear regression model from the sample of the data we have in the training set:

$$\hat{\beta}_0 \ , \ \hat{\beta}_1 \ \dots \ \hat{\beta}_m$$

- The process of estimating these parameters is called “fitting the model”

Estimated linear regression model

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X$$

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + ... + \hat{\beta}_m X_m$$

Y "hat" defines our regression line

Estimating model parameters is pure math

- We estimate the slope first
- Then we estimate the intercept

$$\hat{\beta}_j = \frac{\Sigma_i^n (x_{ji} - \bar{x_j})(y_i - \bar{y})}{\Sigma_i^n (x_{ji} - \bar{x_j})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_j \bar{x}$$

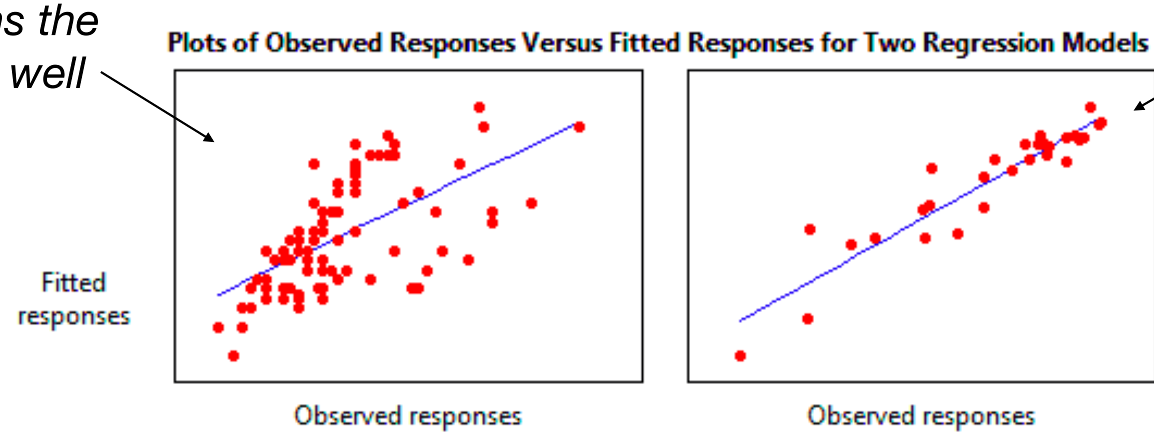
- https://www.youtube.com/watch?v=BLRjywb0mes

Measuring goodness of fit

- We can always fit a line through our data but how well does it fit?
- We measure goodness of regression model fit using
  - Tells us how much of the data is explained by the model

Weak fit = model somewhat explains the data but not very well

Good fit = model explains most of the data very well



https://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit

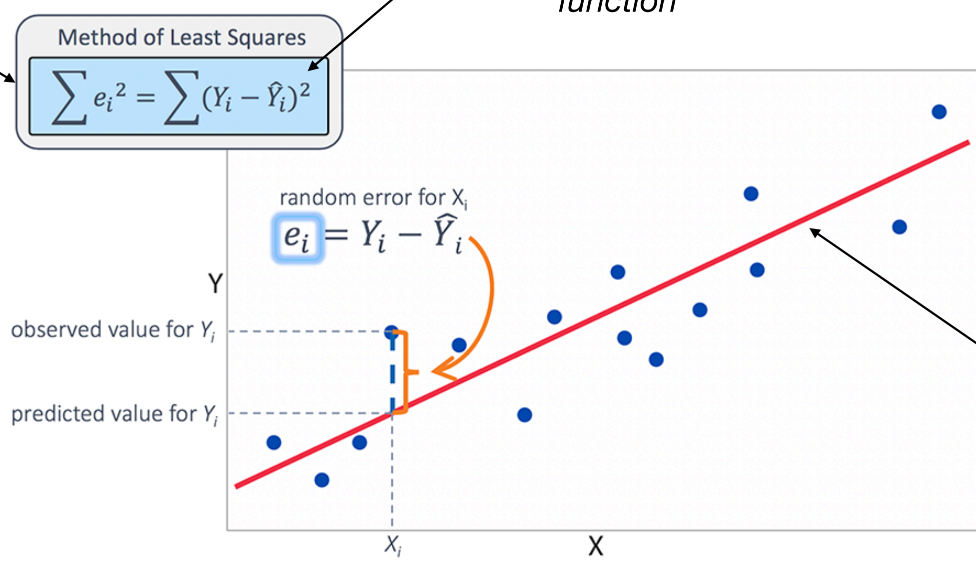
Measuring goodness of fit (cont'd)

- We measure goodness of fit by looking at how different the predictions are from known values of the dependent variable

Least Squares regression (also called Ordinary Least Squares)

Often referred to as cost function

We optimize the model parameters to minimize total differences between the predicted and the true values



Regression line

Note: least absolute errors method aims to minimize the sum of absolute errors

https://www.jmp.com/en_hk/statistics-knowledge-portal/what-is-multiple-regression/fitting-multiple-regression-model.html

Representing least squares linear multiple regression as an optimization problem

Matrix notation:

$$\min_{\beta} (\mathbf{y} - \mathbf{X}\beta)^{\mathsf{T}} (\mathbf{y} - \mathbf{X}\beta)$$

https://en.wikipedia.org

Equation notation:

$$\underset{\beta \in \mathbb{R}}{argmin} \sum [y_i - \hat{y}_i] = \underset{\beta \in \mathbb{R}}{argmin} \sum [y_i - (\beta_0 + \beta_1 x_1 + \beta x_2 + \cdots + \beta x_p)]^2$$

https://towardsdatascience.com/

Optimizing for the model parameters

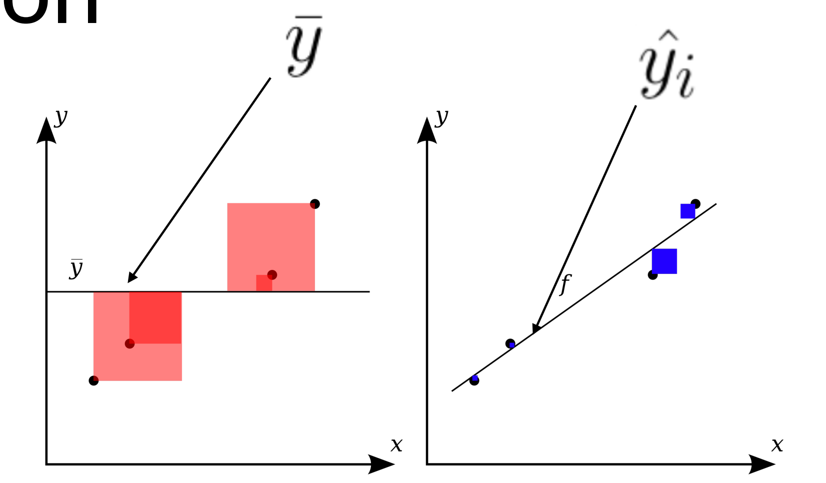
- Take partial derivatives with respect to each Theta parameter and set each equation to zero
- Solve the system of equations to get each Theta
- Thankfully ML libraries already perform this optimization in the background

$$J(\theta_0, \theta_1, \ldots, \theta_m) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Another form we can see the cost function written as

Coefficient of determination

- Referred to as R^2 (R squared)
- Proportion of variance in the model that is explained by the independent variables



https://en.wikipedia.org/

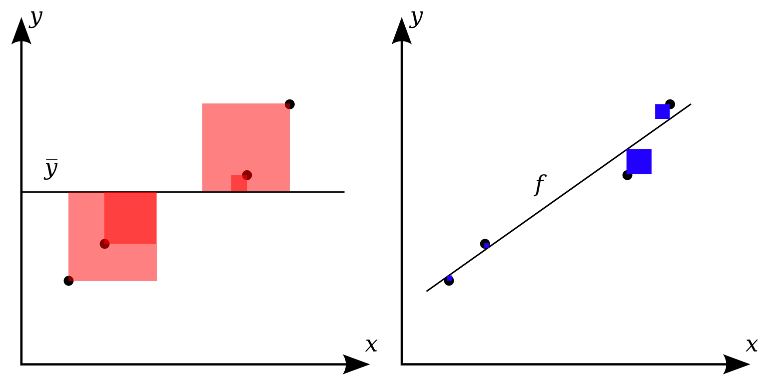
Regression sum of squares $\sum_{i}^{n}(\hat{y}_i - \bar{y})^2$

$$R^2 = \frac{SS_{\text{reg}}}{SS_{\text{tot}}}$$

Total sum of squares $\sum_{i}^{n}(y_i - \bar{y})^2$

Coefficient of determination

- Referred to as R^2 (R squared)
- Proportion of variance in the model that is explained by the independent variables



https://en.wikipedia.org/

Regression sum of squares $\sum_{i}^{n}(\hat{y}_i - \bar{y})^2$

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \qquad R^2 = \frac{SS_{\text{reg}}}{SS_{\text{tot}}}$$

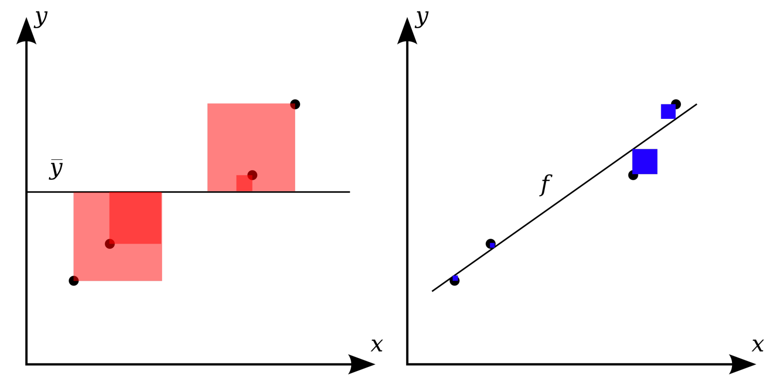
Residual sum of squares: $\sum_{i}^{n}(y_i - \hat{y}_i)^2$

$$SS_{tot} = SS_{res} + SS_{error}$$

Total sum of squares $\sum_{i}^{n}(y_i - \bar{y})^2$

Coefficient of determination

- Referred to as R^2 (R squared)
- Proportion of variance in the model that is explained by the independent variables



https://en.wikipedia.org/

Regression sum of squares $\sum_i^n (\hat{y}_i - \bar{y})^2$

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \qquad R^2 = \frac{SS_{\text{reg}}}{SS_{\text{tot}}}$$

Residual sum of squares: $\sum_i^n (y_i - \hat{y}_i)^2$

$$SS_{tot} = SS_{res} + SS_{error}$$

Total sum of squares $\sum_i^n (y_i - \bar{y})^2$

The goal of model fitting is to minimize this error

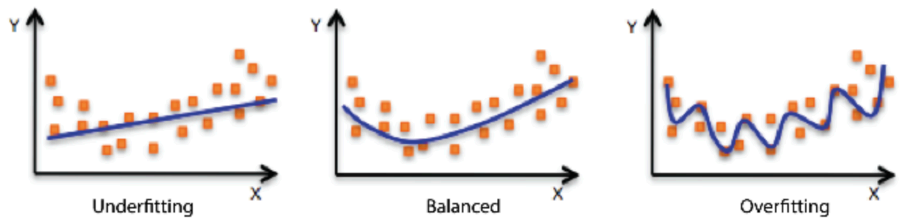
Regularized regression

- The main idea is to introduce a small amount of bias into the model to prevent overfitting to the training data
  - In return get a decrease in model variance
    - Get worse fit to the training data but better performance on the test data
- We add a small penalty to the regression equation (lambda times the slope) $\longrightarrow \lambda\beta$
- Regularization is especially helpful when the number of model parameters greatly exceeds the number of available training observations
  - Often the case in biomedical research
    - E.g. a whole transcriptome sample has ~20,000 individual gene features but a “big” dataset is normally <500 samples
  - Rule of thumb is that you need at least as many training examples as the number of parameters the model contains

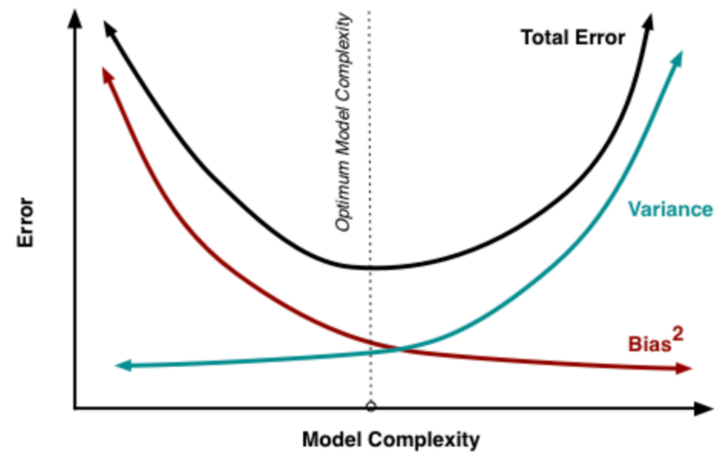
Regularized regression (cont'd)

High bias,
Low variability

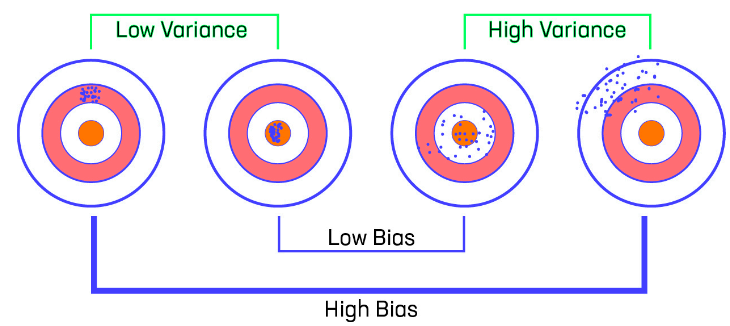
Low bias,
high variability



https://towardsdatascience.com/ridge-regression-for-better-usage-2f19b3a202db



https://towardsdatascience.com/ridge-regression-for-better-usage-2f19b3a202db



www.geeksforgeeks.org

We want our model to fit well both the training and the test sets; we want to achieve both low bias and low variance

Ridge regression (Tikhonov regularization)

- Uses L2 loss/penalty (squared penalty)
  - Called penalty since it increases residual error
- OLS regression models treat all independent variables equally, becomes more complex with more variables, and tends to produce low bias models but does not always produce low variance models
- *Lambda* (regularization parameter) must be tuned for each model (e.g. cross-validation)
  - When lambda is zero, ridge regression reduces to OLS

$$\min_{\beta} (\mathbf{y} - \mathbf{X}\beta)^{\mathsf{T}} (\mathbf{y} - \mathbf{X}\beta) + \lambda(\beta^{\mathsf{T}}\beta - c)$$

Lagrangian optimization function

Lagrange multiplier

https://en.wikipedia.org

$$\hat{\beta}^{ridge} = \underset{\beta \in \mathbb{R}}{argmin} \|y - XB\|_2^2 + \lambda \|B\|_2^2$$

https://towardsdatascience.com/

$$\|B\|_2 = \sqrt{\beta_0^2 + \beta_1^2 + \cdots + \beta_p^2}$$

Vector norm

Lasso regression

- Least Absolute Shrinkage and Selection Operator (LASSO)
- Uses L1 loss/penalty (absolute penalty)
- Can lead to some of the parameters to be estimated to be zero
  - Effectively performs feature selection
  - Handy when we need to select "important" features or when there are highly correlated independent variable in our feature set
- *Lambda* is the regularization parameter and must be tuned for each model (e.g. cross-validation)

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\}$$

https://en.wikipedia.org

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

https://towardsdatascience.com/

Ridge vs. Lasso

- Lasso regression may set coefficients to zero while ridge regression will never set coefficients to zero
- Elastic net attempts to combine both ridge and lasso penalties

$$L_{ridge} = argmin_{\hat{\beta}} \left(\|Y - \beta * X\|^2 + \lambda * \|\beta\|_2^2 \right)$$

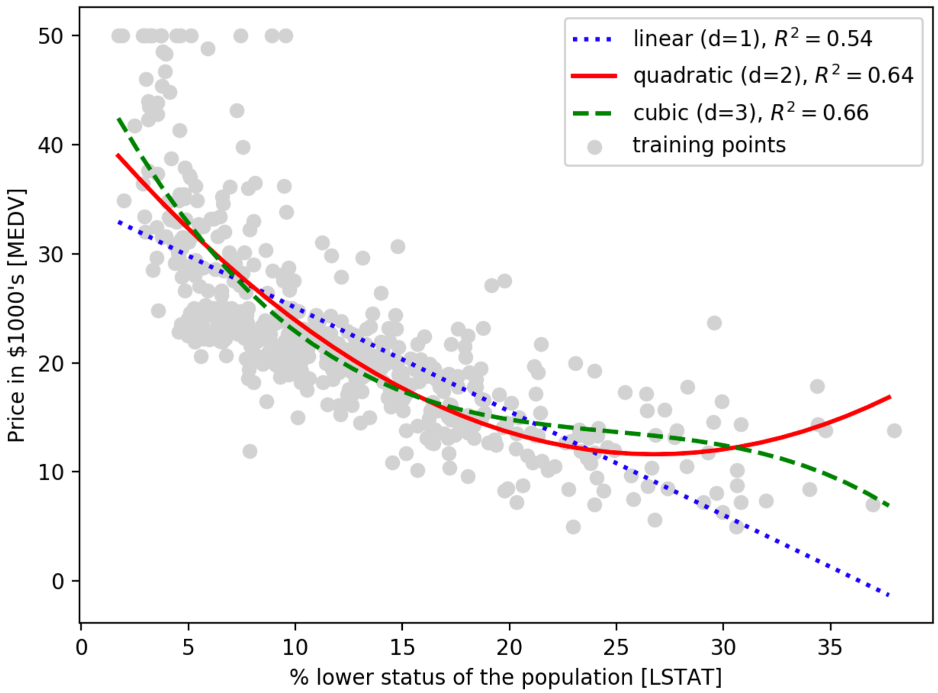
$$L_{lasso} = argmin_{\hat{\beta}} \left(\|Y - \beta * X\|^2 + \lambda * \|\beta\|_1 \right)$$

$$L_{elasticNet} = argmin_{\hat{\beta}} \left(\hat{\beta} \right) \left(\sum \left(y - x_i^J \hat{\beta} \right)^2 \right) /2n + \lambda \left((1 - \alpha)/2 * \sum_{j=1}^{m} \hat{\beta}_j^2 + \alpha * \sum_{j=1}^{m} \left\| \hat{\beta}_j \right\| \right)$$

www.geeksforgeeks.org

Non-linear regression

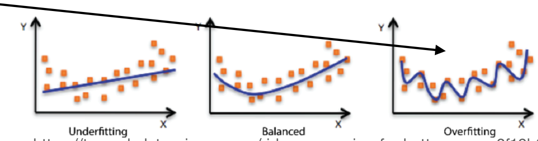
- Data is modeled by a non-linear function
  - Not all data follows a linear relationship pattern
  - Sometimes you need to play around with the data to figure out which type of model fits it best



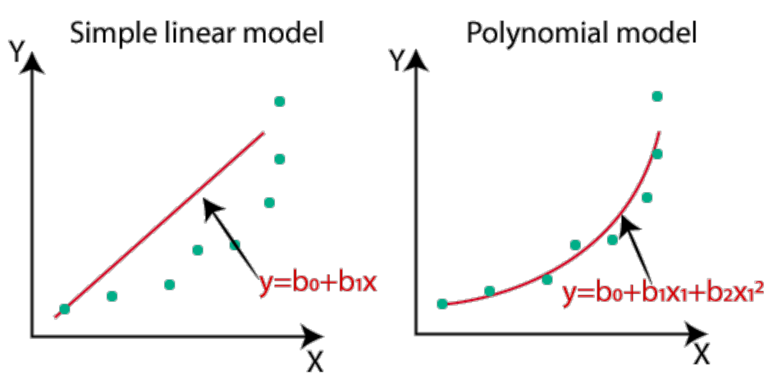
https://charleshsliao.wordpress.com/2017/06/16/ransac-and-nonlinear-regression-in-python/

Polynomial regression

- This type of regression creates additional variables that are powers of the original variables
  - Can be thought of as feature engineering
- Any degree polynomial can be inserted into the model
- Be aware that using higher degree polynomials can cause overfitting



https://towardsdatascience.com/ridge-regression-for-better-usage-2f19b3a202db



javatpoint.com

Types of Polynomials

Linear —— $ax + b = 0$

Quadratic —— $ax^2 + bx + c = 0$

Cubic —— $ax^3 + bx^2 + cx + d = 0$

https://medium.com

Why we actually call it polynomial ***linear*** regression?

- The term "linear" refers to the linear combination of the parameters/ coefficients of the model
  - Coefficients are linear
- We are not talking about the independent variables themselves and how they are treated
  - In polynomial linear regression independent variables are non-linear

Simple Linear Regression	$y = b_0 + b_1 x_1$
Multiple Linear Regression	$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$
Polynomial Linear Regression	$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$

https://medium.com

Polynomial regression model in matrix form

- General population polynomial regression model can be presented in a matrix form as follows:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

https://en.wikipedia.org

$$\widehat{\vec{\beta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\ \mathbf{X}^\mathsf{T}\vec{y}$$

https://en.wikipedia.org

X is a Vandermonde matrix
(matrix of geometric progressions)

We can generalize this model to a multivariate polynomial linear regression model

- We can also treat relationships between different independent variables and the dependent variable differently
  - Maybe there is a linear relationship between X1 and Y but quadratic relationship between X2 and Y

Assumptions of polynomial regression

- The additive relationships between the dependent variable and a set of independent variables in the population can be explained by a polynomial line
- Independent variables are independent of each other
- Errors associated with each independent variable are also independent of each other

How do we find the correct degree of polynomial?

- Like in other cases of hyperparameter tuning you try a number of different models and pick the one that performs the best in cross-validation experiment
  - Forward selection - starting with a linear regression and increasing the degree of polynomial up
  - Backward selection - starting with a high degree and going down to linear function

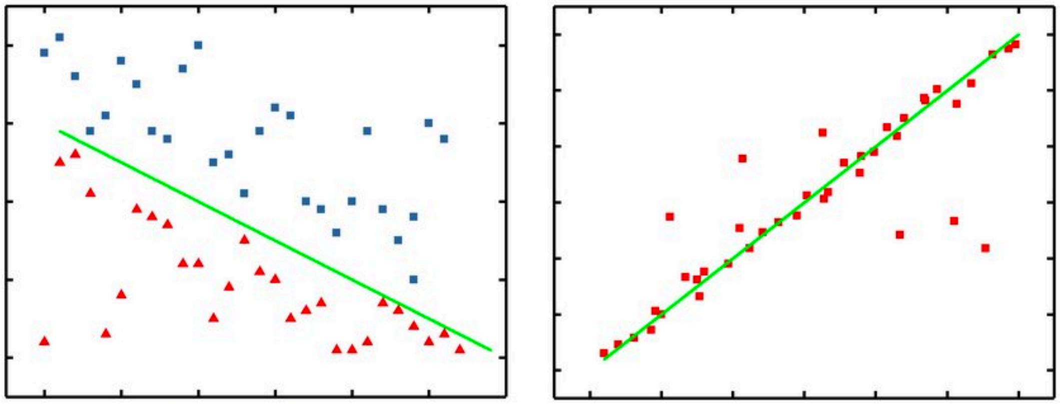
Logistic regression

- A regression analysis can be used for classification problems
  - Usually by converting the prediction to a probability or odds of the observation belonging to a certain class
- Logistic regression models output the probability of an event/class based on the values of the input independent variables
  - Input variables can be numeric or categorical
- Performs classification based on the estimated probability that an observation is of a given class
- Goal: convert predicted real values to categories (e.g. 0 or 1 for binary classification)

Types of logistic regression

- Binary logistic regression - two class classification problem
  - E.g. disease vs. not
- Multinomial logistic regression - two or more class classification problem
  - E.g. type of animal: cat, dog, horse, spider
- Ordinal logistic regression - two or more class with ordering classification problem
  - E.g. low, medium, high

Logistic regression vs. linear regression



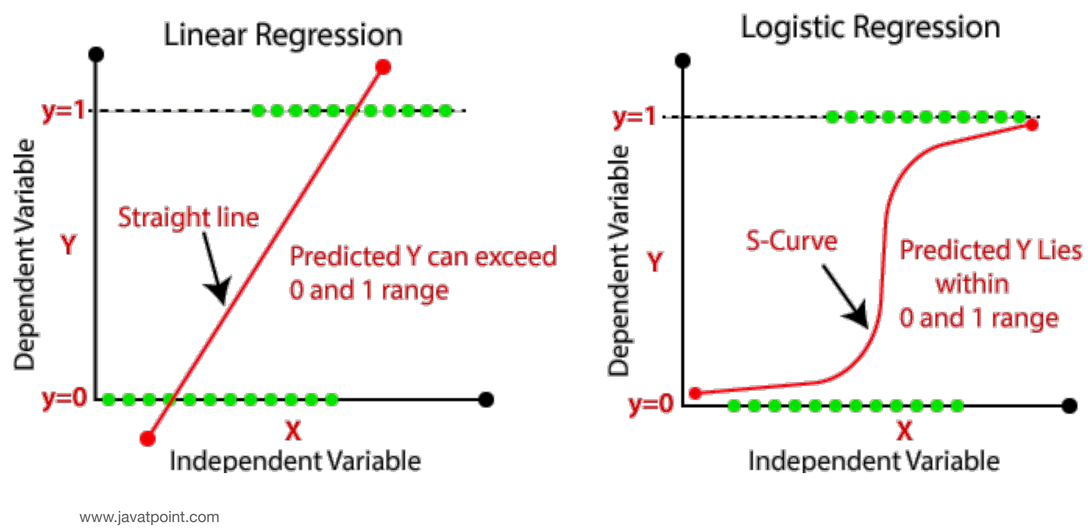
(a) Logistic Regression (b) Linear Regression

"Regression Analysis With Differential Privacy Preserving" IEEE Access 2019

Why linear regression does not work for classification problems

- Predicted output values are not categorical and can range from negative infinity to positive infinity
  - Not confined by the range [0,1] for probabilities
  - Finding the right threshold for separating classes within the output value can be hard
  - And how do we deal with multi-class problems?
- Categorical data (predicted class) is not normally distributed
- Probabilities themselves are often not non-uniformly distributed across all values of independent variables

Why linear regression does not work for classification problems (cont'd)



Quick review of probabilities and odds

$$P = \frac{outcomes\ of\ interest}{of\ all\ outcomes}$$

Probability

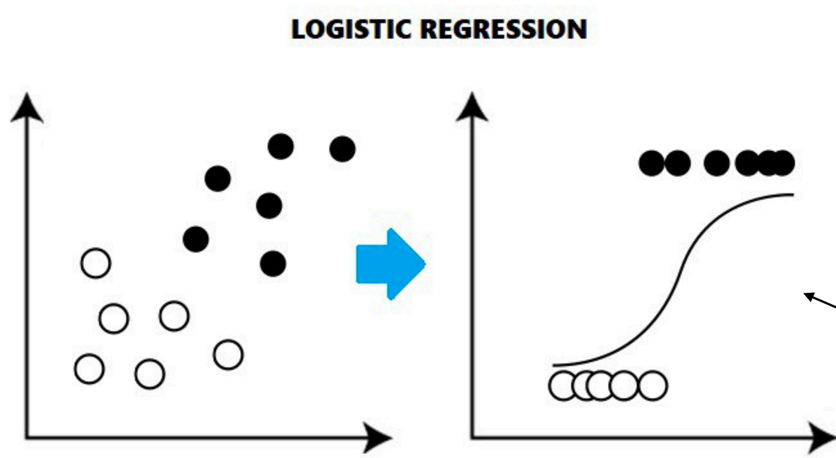
$$odds = \frac{P(outcome\ occurring)}{P(outcome\ not\ occurring)}$$

Odds

$$odds = \frac{P}{1-P}$$

$$odds\ ratio = \frac{odds(outcome1)}{odds(outcome2)}$$

Turning a continuous output into a probability

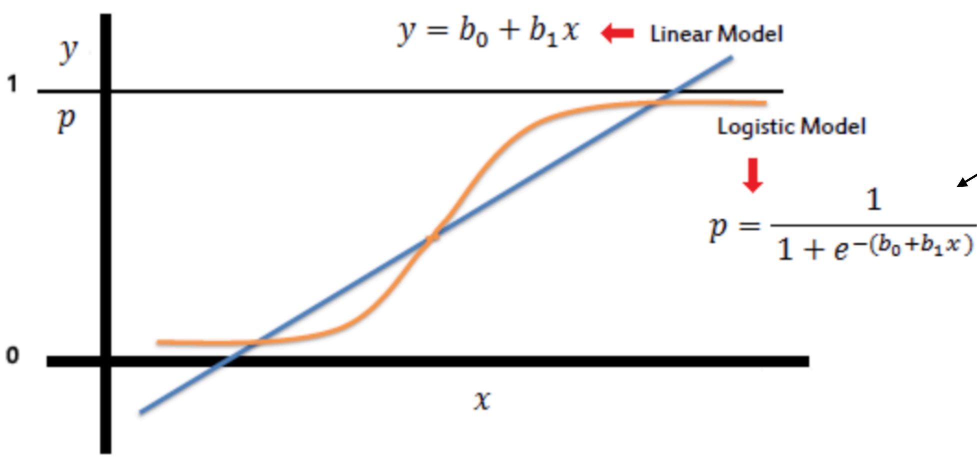


https://medium.com

We use logistic functions to do this

Logistic regression model

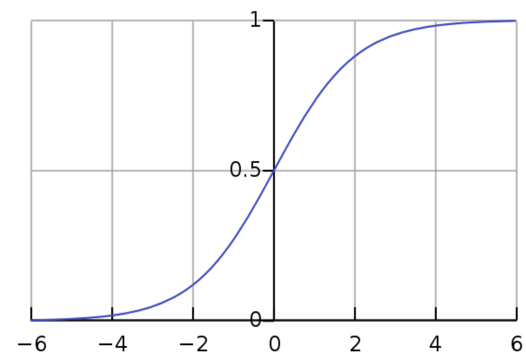
- Logistic function gives us ability to convert predicted value into a probability



https://medium.com

Often called sigmoid function

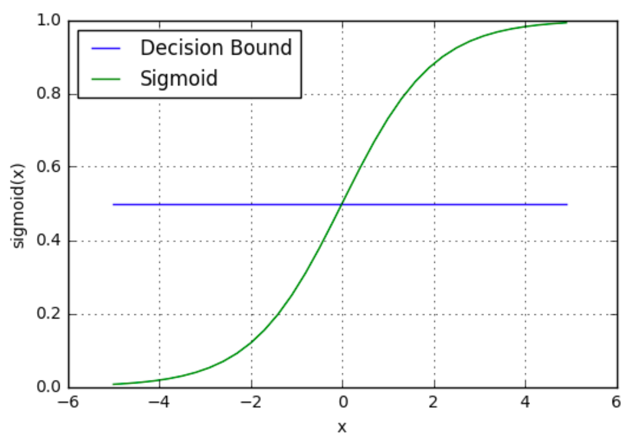
Logistic function $f(x) = \frac{L}{1+e^{-k(x-x_0)}}$



https://en.wikipedia.org/

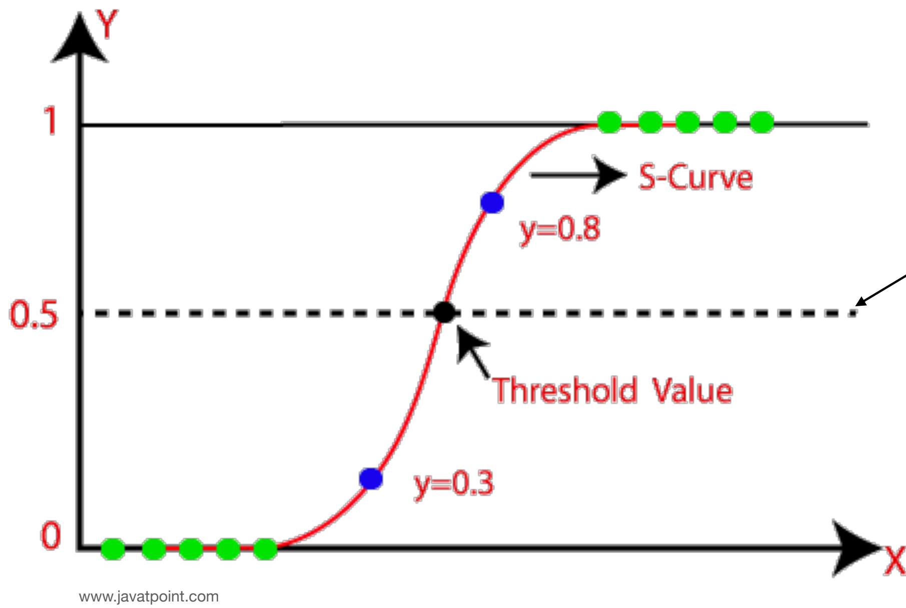
Classification with logistic regression

- We can estimate class based on the dependent variable estimate



ml-cheatsheet.readthedocs.io

Generally the threshold is at P = 0.5



Decision hyperplane

The further away the data point is from the decision hyperplane, the more confident the model is about this point's class assignment (indicated by the higher probability)

Why logistic function?

- Let’s say we want to understand log odds of event occurring (the observation belongs to class A)
- We can present log odds as a linear model of the estimated parameters and the input data:

$$\ell = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

A model with two predictor/independent variables

Exponentiate to recover odds ratio

$$\frac{p}{1-p} = b^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}$$

Algebraic manipulations

$$p = \frac{b^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}}{b^{\beta_0 + \beta_1 x_1 + \beta_2 x_2} + 1} = \frac{1}{1 + b^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}}$$

Logistic function

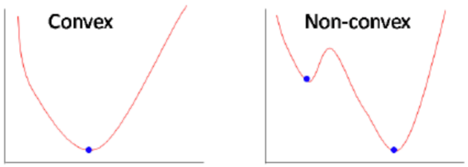
https://en.wikipedia.org/

Interpretation of logistic regression coefficients

- Intercept is the value of log odds when no information is used to evaluate the prediction (random classifier)
  - Intercept correction is a procedure used to “correct” a classifier that was trained on unbalanced class data that does not reflect population class proportions
    - E.g. see “Logistic Regression in Rare Events Data” by Gary King and Langche Zeng, Political Analysis 2001
- Each slope coefficient reflects the amount of change in log odds with each additional independent variable
  - Can see how much each variable contributes to the classification

Logistic regression cost function

- We cannot use the same cost function (quadratic loss) as in linear regression
  - Applying linear regression loss to classification problems will produce non-convex function shape
    - Because sigmoid function is non-linear
    - The function shape will have many local minima - hard to optimize with gradient descent



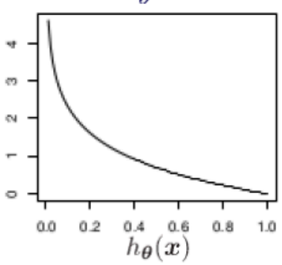
Left(Linear Regression mean square loss), Right(Logistic regression mean square loss function)

https://medium.com

Logistic regression cost function (cont'd)

$$\text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}),\ y) = \begin{cases} -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } \ y = 1 \\ -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } \ y = 0 \end{cases}$$

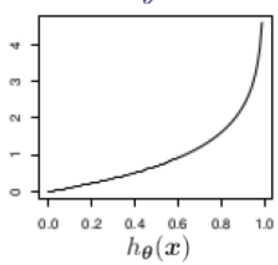
if $\ y = 1$



if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 1$
then cost $= 0$

if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \to 0$
then cost $\to \infty$
predicted
prob$(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 0$
but $y = 1$

if $\ y = 0$



if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0$
then cost $= 0$

if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \to 1$
then cost $\to \infty$
predicted
prob$(y = 0 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 0$
but $y = 0$

Combining two cost functions into one

Parameters of the model:

$$\frac{1}{1 + b^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}}$$

$$\text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}),\ y) = -y \log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) - (1 - y) \log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}))$$

if $y = 1$: $\quad \text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x}))$

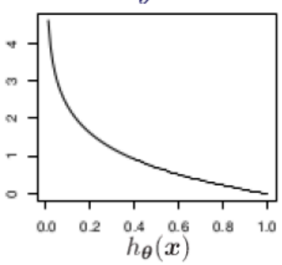
if $y = 0$: $\quad \text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}))$

https://medium.com

Logistic regression cost function (cont’d)

$$\text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}),\ y) = \begin{cases} -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if}\ \ y = 1 \\ -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if}\ \ y = 0 \end{cases}$$

if $y = 1$

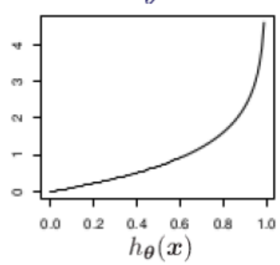


if $h_{\theta}(x) = 1$ then cost $= 0$

if $h_{\theta}(x) \to 0$ then cost $\to \infty$

predicted prob$(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 0$ but $y = 1$

if $y = 0$



if $h_{\theta}(x) = 0$ then cost $= 0$

if $h_{\theta}(x) \to 1$ then cost $\to \infty$

predicted prob$(y = 0 \mid \boldsymbol{x}; \boldsymbol{\theta}) = 0$ but $y = 0$

https://medium.com

Combining two cost functions into one

Parameters of the model:

$$\frac{1}{1 + b^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}}$$

$$\text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}),\ y) = -y \log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) - (1 - y)\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}))$$

if $y = 1$: $\quad \text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x}))$

if $y = 0$: $\quad \text{cost}\,(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}))$

We optimize the model parameters across all training samples by minimizing the loss function:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1 - y^{(i)})\log(1 - h_\theta(x^{(i)}))]$$

Evaluating a logistic regression model

- Goodness of fit
  - R^2 (coefficient of determination)
  - Akaike Information Criteria (AIC)
- Other ways we discussed in the introduction lecture
  - Balanced accuracy
  - Confusion matrix
  - AUC
- Other methods are being used as well

Linear vs. logistic regression

Linear Regression	Logistic Regression
Linear regression is used to predict the continuous dependent variable using a given set of independent variables.	Logistic Regression is used to predict the categorical dependent variable using a given set of independent variables.
Linear Regression is used for solving Regression problem.	Logistic regression is used for solving Classification problems.
In Linear regression, we predict the value of continuous variables.	In logistic Regression, we predict the values of categorical variables.
In linear regression, we find the best fit line, by which we can easily predict the output.	In Logistic Regression, we find the S-curve by which we can classify the samples.
Least square estimation method is used for estimation of accuracy.	Maximum likelihood estimation method is used for estimation of accuracy.
The output for Linear Regression must be a continuous value, such as price, age, etc.	The output of Logistic Regression must be a Categorical value such as 0 or 1, Yes or No, etc.
In Linear regression, it is required that relationship between dependent variable and independent variable must be linear.	In Logistic regression, it is not required to have the linear relationship between the dependent and independent variable.
In linear regression, there may be collinearity between the independent variables.	In logistic regression, there should not be collinearity between the independent variable.

www.javatpoint.com

Generalizing to multiple logistic regression model

- We can generalize the simple logistic regression model to multiple logistic regression model (a model with multiple independent variables)
  - We now optimize for more slopes but the concepts are the same

$$\pi(x) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p}}$$

Activation function

- We have been talking about logistic function
  - Also called sigmoid function
- Sigmoid function is an activation function for logistic regression
  - An activation function performs non-linear transformation of the input vector
- We will re-visit the concept of activation functions when we get to neural networks

Multi-class logistic regression

- We now have more than 2 classes so [0,1] classification will not work
  - In binary classification we have y = [0, 1]
  - In multi-class classification we have y = [0,1,…,k]
    - k-class problem
- We use softmax activation
  - Softargmax or normalized exponential function (softmax function)
    - Takes in a vector of real numbers and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input
      - These probabilities sum up to 1

Softmax function

- Softmax function is also a type of an activation function
  - Transforms the output of the regression model into probabilities
- We will see it again when we talk about neural networks

$$\sigma(z_i) = \frac{e^{z_{(i)}}}{\sum_{j=1}^{K} e^{z_{(j)}}} \quad for\ i = 1, ., ., ., ., K\ and\ z = z_1, ., ., ., ., z_K$$

Using regularization with logistic regression

- Regularization can be used with logistic regression in the same way we use it with linear regression

ridge regression

$$J(\mathbf{w}) = \sum_{i=1}^{n} \left[-y^{(i)} \log\left(\phi(z^{(i)})\right) - \left(1 - y^{(i)}\right) \log\left(1 - \phi(z^{(i)})\right) \right] + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

kdnuggets.com

Some concluding remarks

- Both linear and logistic regression models are generalized linear models (GLM)

- Let’s look at some python code examples in Jupyter notebooks
  - *Regression.Iris.ipynb*
  - *Perceptron.Breast.ipynb*
  - *Regression.Boston.ipynb*