Data Science

Yogesh Kulkarni

Linear Regression

Want to know future? Predictions

- How does sales volume change with changes in price.
- How is this affected by changes in the weather?
- How does the amount of a drug absorbed vary with dosage and with body weight of patient?
- Does it depend on blood pressure?

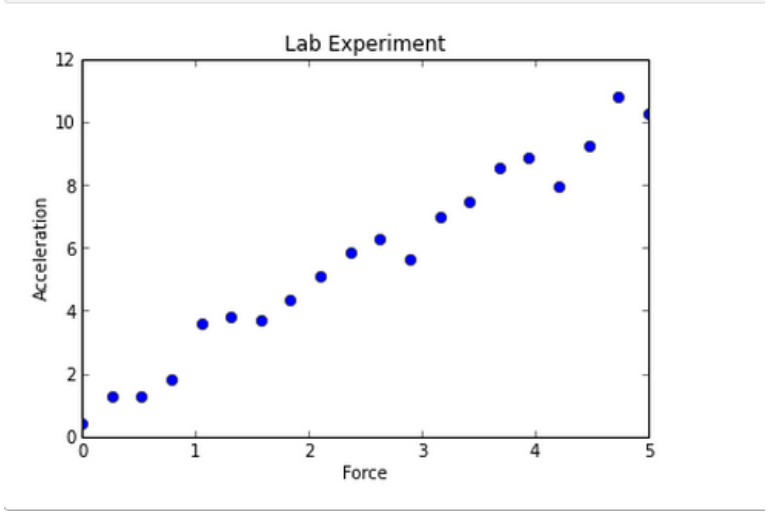
Answering questions like these, requires us to create a model.

The model

- A model is a formula where one variable (response) varies depending on one or more independent variables (covariates).
- The formula (model) can be simple or complex.
- Simplest is a Linear Model where the dependent varies linearly with the independent variable(s).
- Creating a Linear Model involves a technique known as Linear Regression.

Remember? - High School Physics Lab

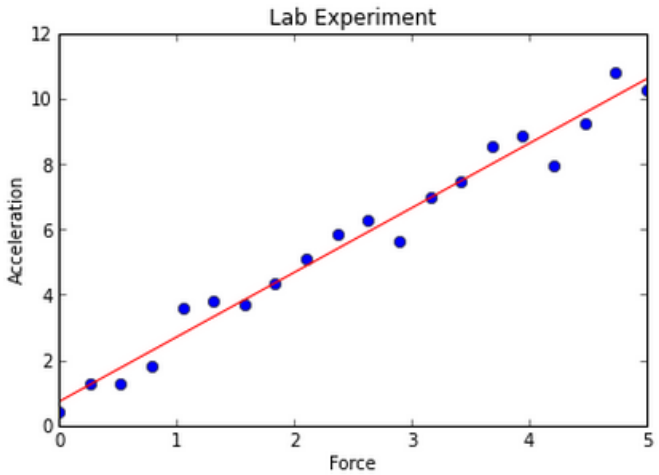
- Finding how acceleration changes when Force is applied
- For some input X (say force), got output Y (say acceleration).
- Plotted the pairs of observations x, y.



What next?

Fitting line

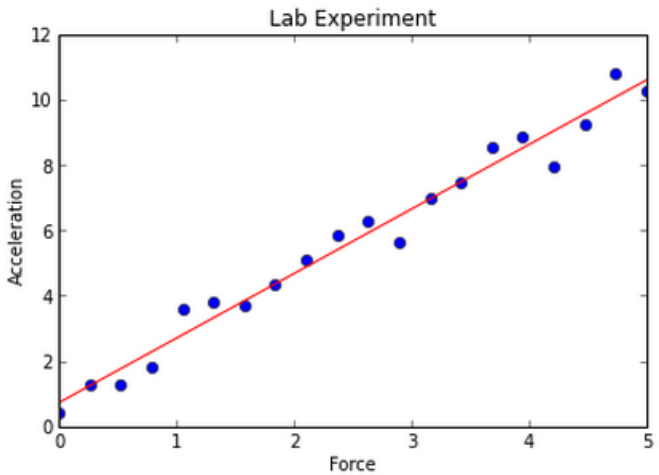
- Then you had to fit a straight line through points
- A visual “best fit”, some points above, some below.
- Found’m’ the slope (how?), and ’b’ (how?).



Whats the model here?

Fitting line

- Equation of line, the formula, itself is the model.
- You were doing informal Linear Regression.



Linear Regression

- What: for predicting a quantitative variable
- Age: "it's been around for a long time". The term regression was devised by Francis Galton in his article Regression Towards Mediocrity in Hereditary Stature in 1886.
- Complexity: easy to understand and use
- Popularity: still widely used

Possible Techniques

- Statisticians know Regression as OLS (Ordinary Least Square) method to fit a line. Deals with minimization of RMS (Root Mean Square) error.
- Theoretical/Analytical Solution $W = (X^T X)^{-1} X^T y$. Not practical for large matrices. Also, matrix is not invertible if any two rows are same.
- Machine Learning Engineers know Regression as prediction of continuous variables (floats). It uses Gradient Descent for the minimization part.

Linear Regression - Analytical Approach (OLS)

Example: Tip Amount

- In a restaurant, tip you pay to the waiter,
- Typically depends on the bill amount.
- Smaller the bill, lesser the tip, and vice versa.

Can we predict the tip amount just by looking at the bill amount?
Inputs?
Outputs?

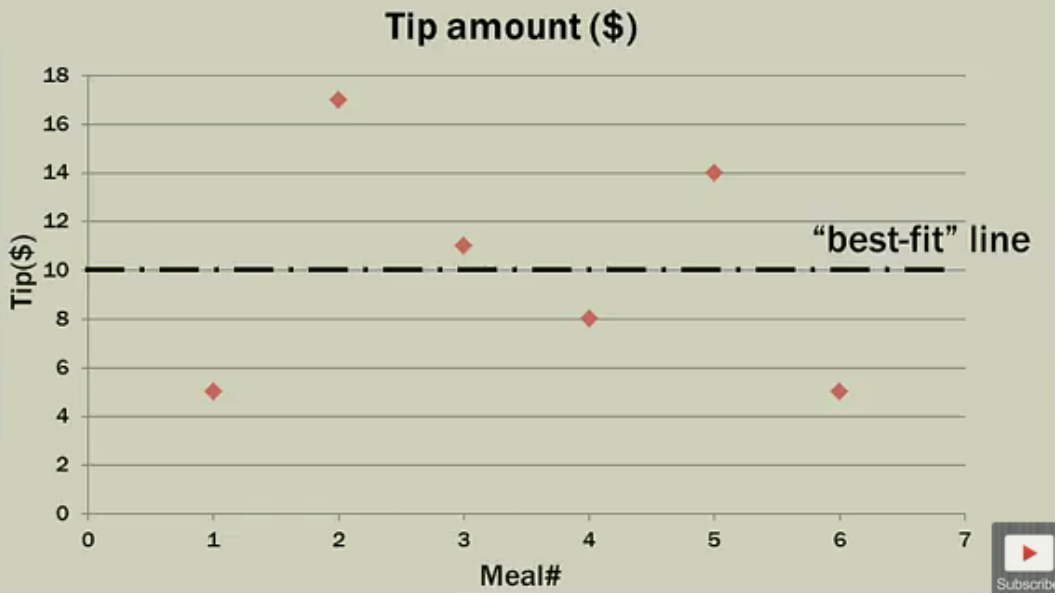
Example: Tip Amount

- You collect, random (meaning, any) 6 tips.
- YOU HAVE NOT COLLECTED THE BILL AMOUNT.
- Just one variable (Meal id is just the descriptor its not a independent variable)
- Any guesses?

Most Basic Situation

Meal#	Tip amount ($)
1	5.00
2	17.00
3	11.00
4	8.00
5	14.00
6	5.00

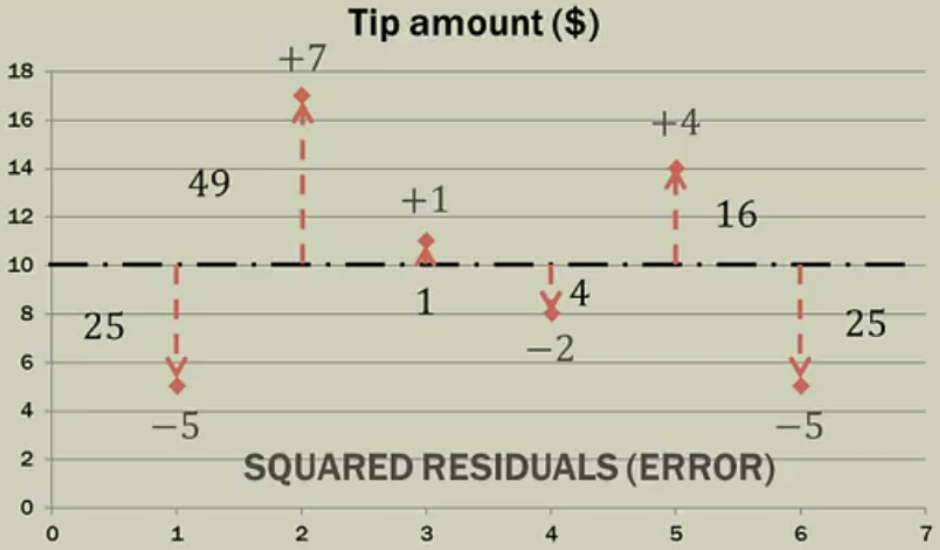
$\bar{y} = \$10$



- With just one variable, mean is the best we can do.
- Note: not a single point falls on it.
- Its just an estimate.
- Goodness of fit?

(Reference: Simple Linear Regression: Step-By-Step - Dan Wellisch)

Example: Tip Amount

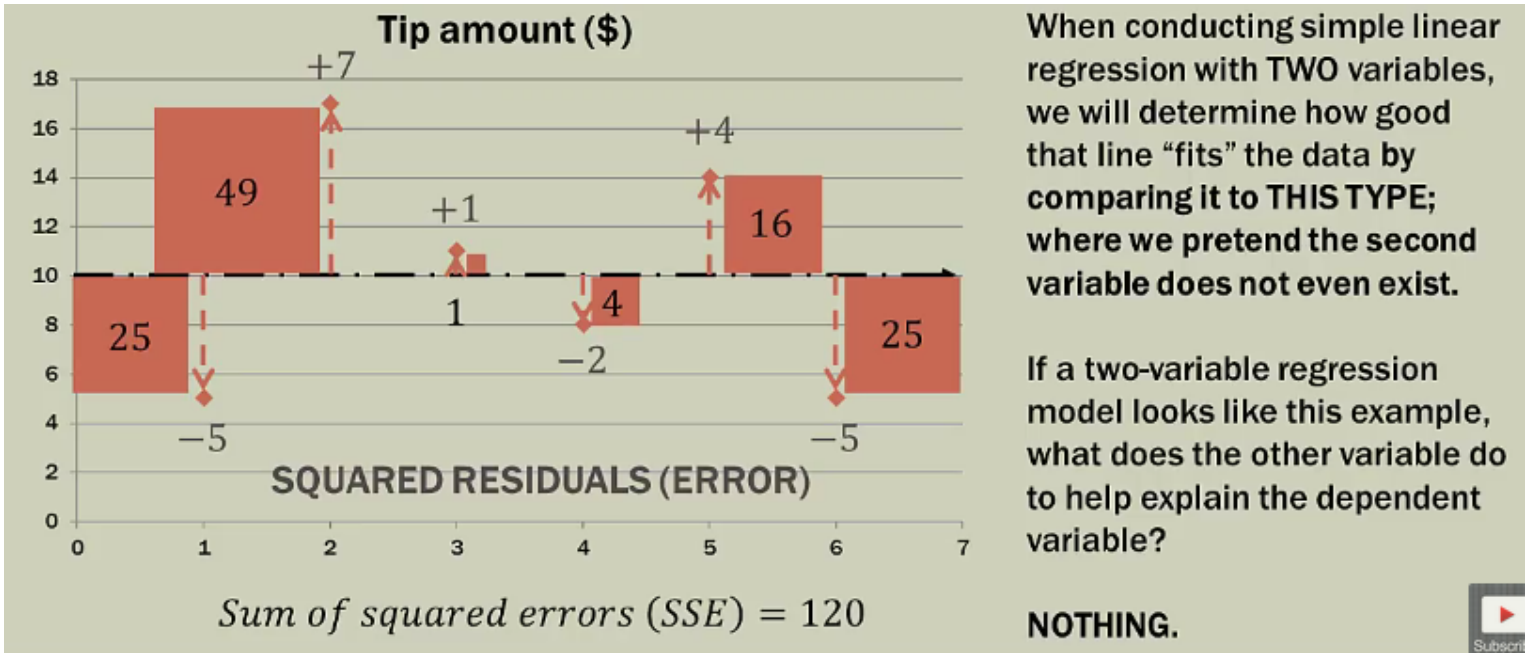


Meal#	Residual	Residual^2
1	−5	25
2	+7	49
3	+1	1
4	−2	4
5	+4	16
6	−5	25

$Sum\ of\ squared\ errors\ (SSE) = 120$

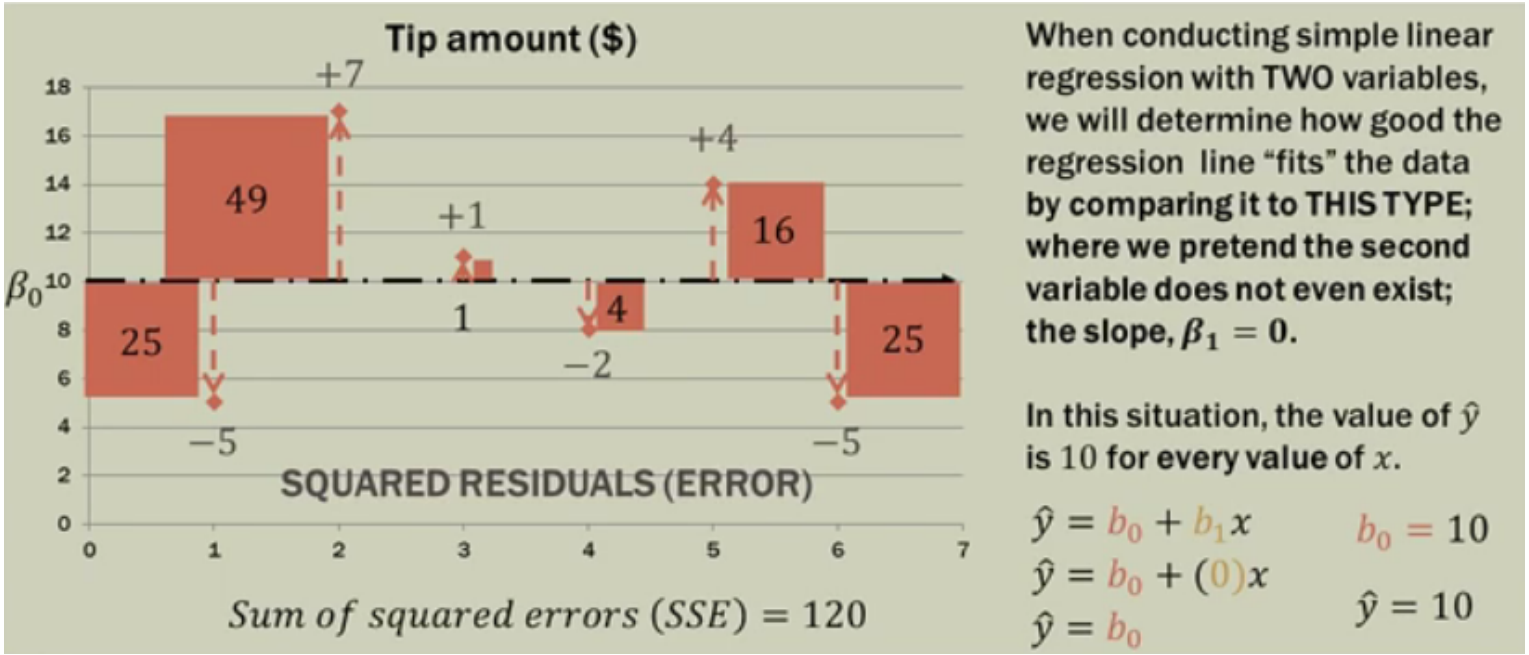
- Std Variation says about goodness of fit.
- Diff of each point with mean are called ‘residuals’ or ‘errors’
- Idea is to minimize SSE (Sum of Squared Errors) or RSS (Residual Sum of Squares)

Important Concept



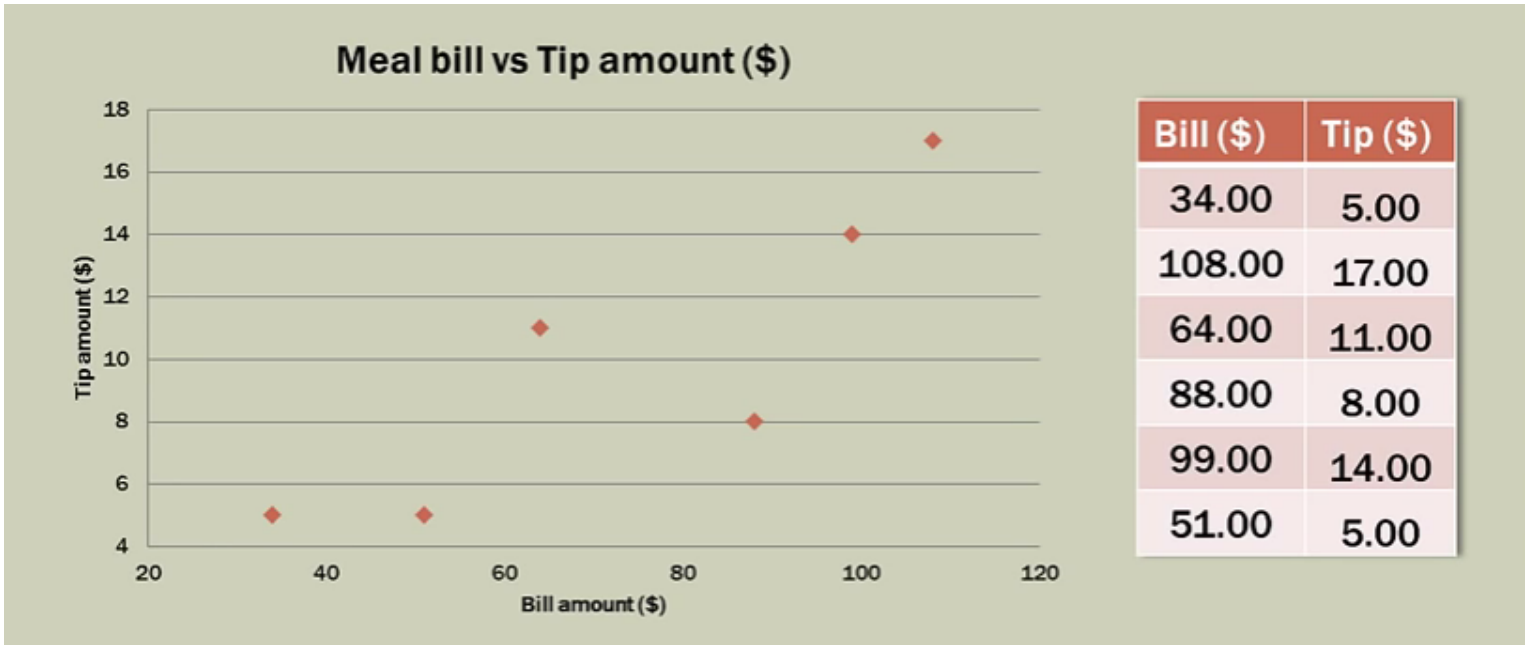
- With just the target variable (tips) we got SSE as 120.
- If we add one more variable to the problem, ie bill amount, what do you expect of SSE?
- It should reduce, or else, whats the use of adding this variable.
- So, we will compare our Linear regression fit line with this (one variable) line.

Important Concept



- We find slope and intercept as b_1 and b_0 respectively.
- A line with $b_1 = 0$ is the horizontal line.
- Its SSE becomes the benchmark. Any selected pair of b_1 and b_0 should give SSE less than the benchmark SSE.

Linear Regression with Two Variables

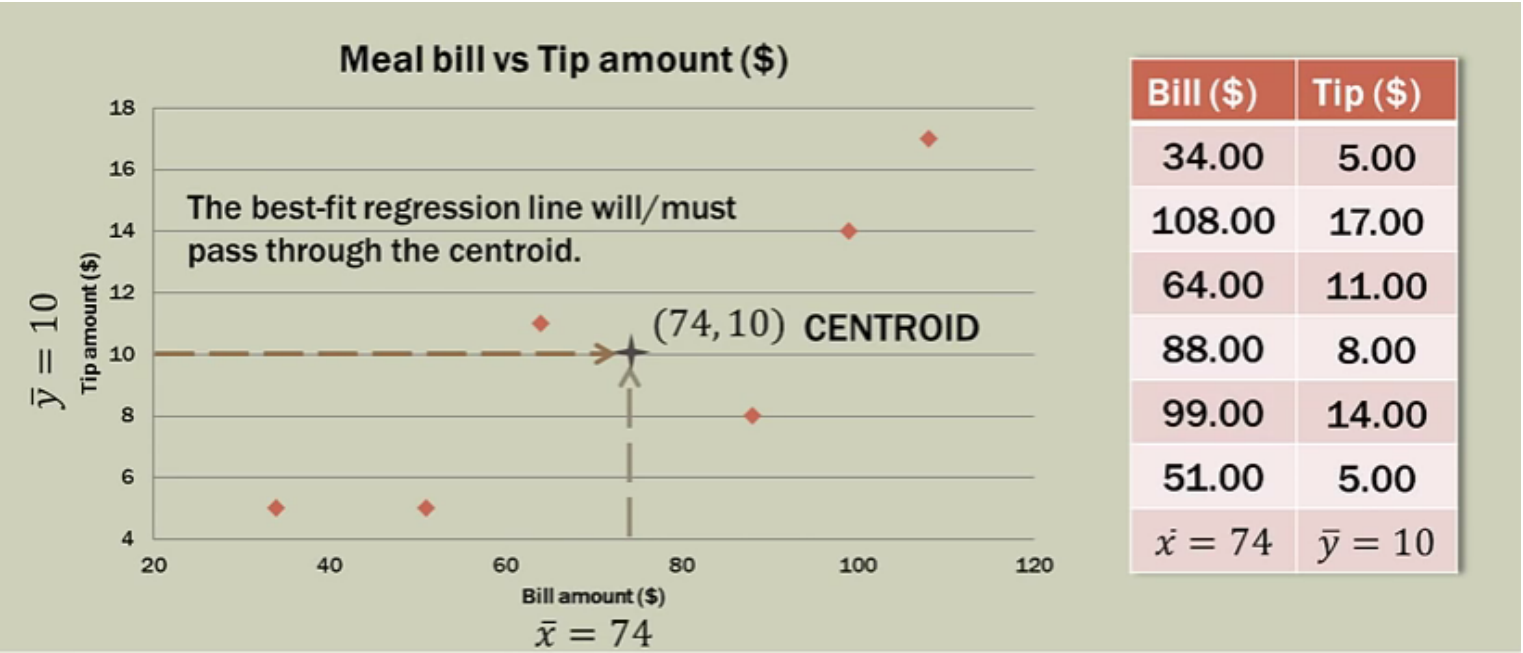


Bill ($)	Tip ($)
34.00	5.00
108.00	17.00
64.00	11.00
88.00	8.00
99.00	14.00
51.00	5.00

- You can fit a line, however you want.
- Even horizontal line can be put
- But then, which one is the best?
- Need to be at least better than horizontal line!!

Methodology

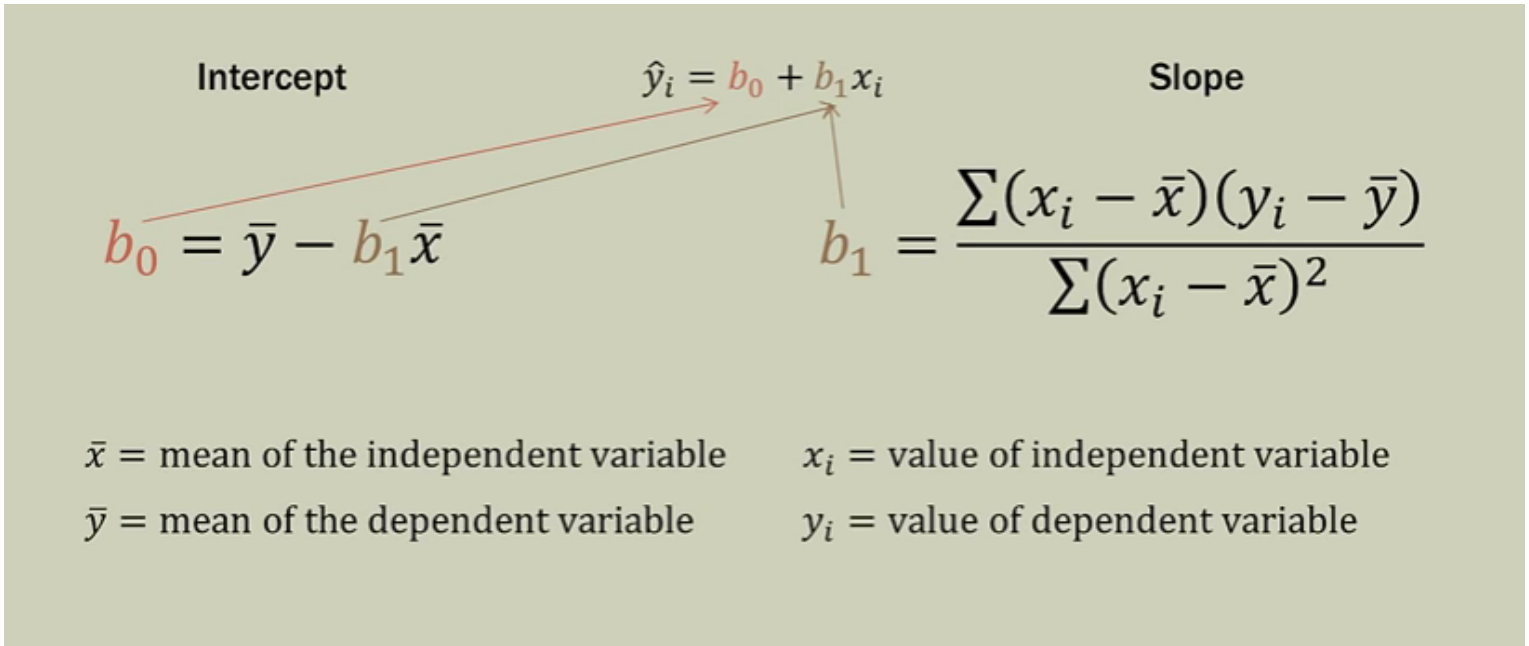
Hypothesis: A line passing through the centroid of the data points will be good. Lets check.



Bill ($)	Tip ($)
34.00	5.00
108.00	17.00
64.00	11.00
88.00	8.00
99.00	14.00
51.00	5.00
$\bar{x} = 74$	$\bar{y} = 10$

- Calculate mean of both x and y.
- Line needs to pass through this point

Calculations



- Intercept and Slope are calculated as above.
- A simple tabular calculations can be done.

Calculations

Meal	Total bill ($)	Tip amount ($)	Bill deviation	Tip Deviations	Deviation Products	Bill Deviations Squared
	x	y	$x_i - \bar{x}$	$y_i - \bar{y}$	$(x_i - \bar{x})(y_i - \bar{y})$	$(x_i - \bar{x})^2$
1	34	5	-40	-5	200	1600
2	108	17	34	7	238	1156
3	64	11	-10	1	-10	100
4	88	8	14	-2	-28	196
5	99	14	25	4	100	625
6	51	5	-23	-5	115	529
	$\bar{x} = 74$	$\bar{y} = 10$			$\sum = 615$	$\sum = 4206$

- Slope comes to $= 615/4206 = 0.1462$
- Intercept comes to $= 10 - 0.1462(74) = -0.818$
- So, eq is $y = 0.1462x - 0.8188$

Interpretation

$$\hat{y}_i = 0.1462x - 0.8188$$

For every \$1 the bill amount (x) increases, we would expect the tip amount to increase by \$0.1462 or about 15-cents.

If the bill amount (x) is zero, then the expected/predicted tip amount is \$-0.8188 or negative 82-cents! Does this make sense? NO. The intercept may or may not make sense in the "real world."

- Slope gives proportionality with which tip is calculated.
- Intercept may or may not make sense in real life.
- Is this manual calculation giving good line?

Evaluation

Meal	Total bill ($)	Observed tip amount ($)	$\hat{y}_i$ (predicted tip amount)	Error $(y - \hat{y}_i)$	Squared Error $(y - \hat{y}_i)^2$
	x	y			
1	34	5	4.1505	0.8495	0.7217
2	108	17	14.9693	2.0307	4.1237
3	64	11	8.5365	2.4635	6.0688
4	88	8	12.0453	−4.0453	16.3645
5	99	14	13.6535	0.3465	0.1201
6	51	5	6.6359	−1.6359	2.6762
	$\bar{x} = 74$	$\bar{y} = 10$		SSE=	$\sum = 30.075$

- Sum of Squares of Error , SSE is 30 for our manual model
- Its far less than the horizontal line benchmark of 120.
- So, by adding Bill amount as independent variable, reduced error by about 90.

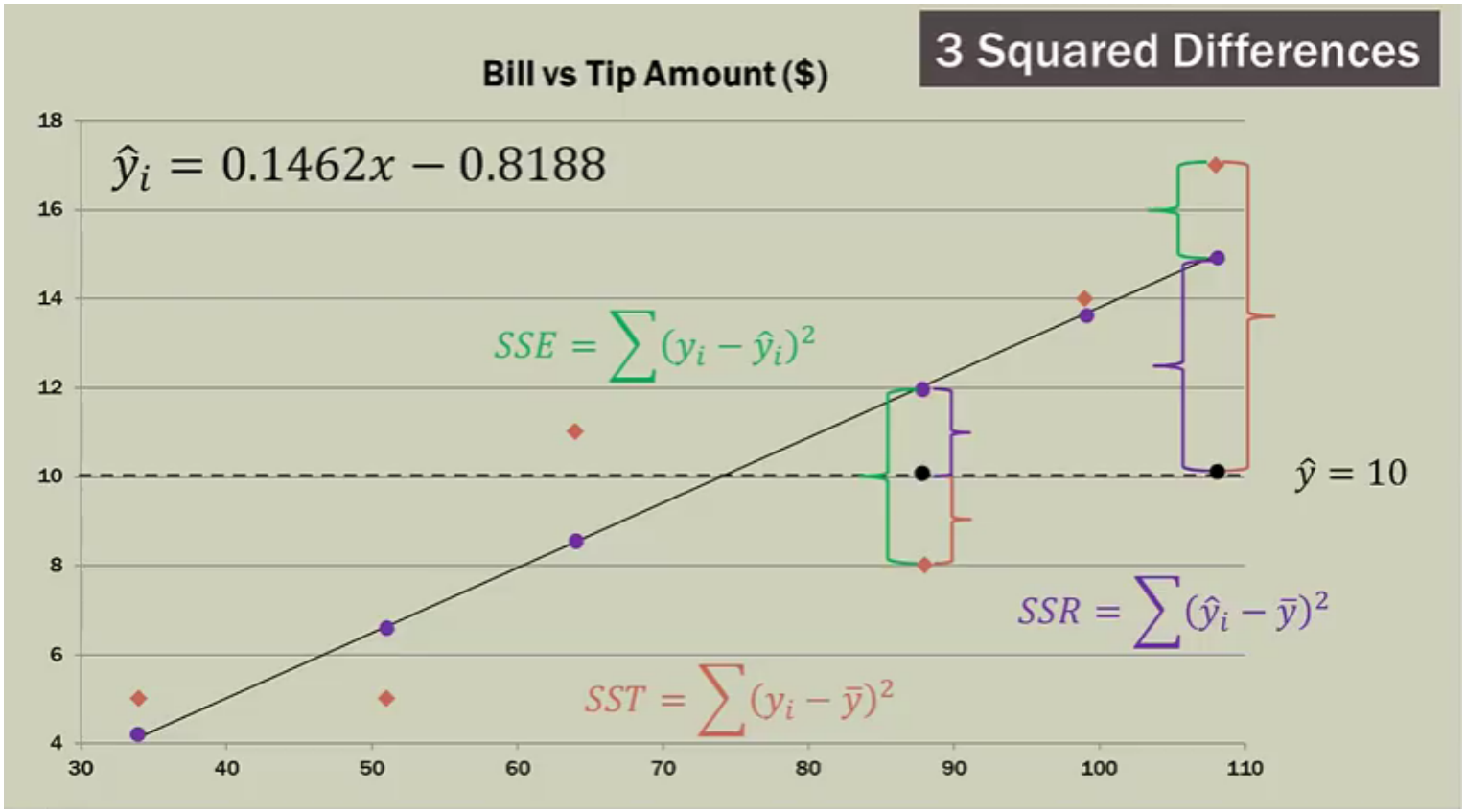
Evaluation

If SSR is large, it uses up more of SST and therefor SSE is smaller relative to SST. The coefficient of determination quantifies this ratio as a percentage.

$$Coefficient\ of\ Determination = r^2 = \frac{SSR}{SST}$$

- SST (sum of squares total) is the benchmark error. Its constant ie 120.
- SSE (sum of squares error) is the level to which we could get error to it by applying regression ie 30.
- The difference, the achievement by regression is called SSR (sum of squares by regression) ie 90.

Evaluation



- For each x, there is predicted y and actual y
- 3 relationships: observed, predicted and total
- Can decide quality based on the ratios among them.

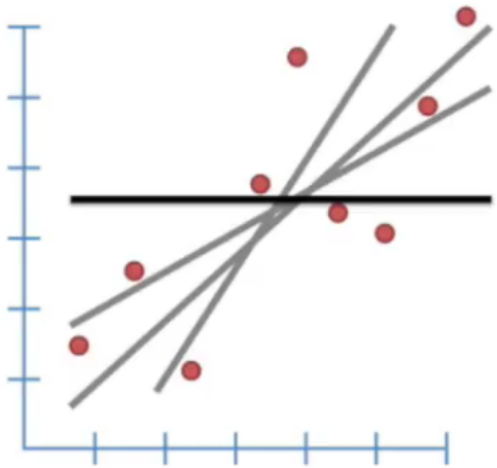
R^2 Statistics

- $SST = SSR + SSE$
- SST: Total variance in the response Y. Amount of variability inherent in the response, before the regression is performed. y_i to $\bar{y}$
- It has two components, one explained by this regression model, other the unexplained.
- SSR: The amount of variability that is explained after performing the regression. $\hat{y}_i$ to $\bar{y}$
- SSE : The amount of variability that is left unexplained, that our model could not minimize. $y_i$to $\hat{y}_i$

Linear Regression - Machine Learning Approach

Linear regression

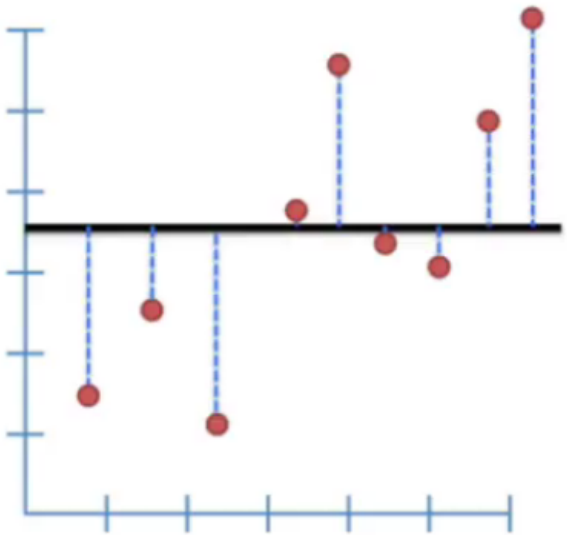
- We have multiple observations of inputs and an output
- Object: to find a model, ie a formulation, adhering to more or less all data points
- ie. to attempt to find the “best” line
- But, which line is the “best”?
- The horizontal line is the worst, the benchmark. We need to beat that!!



(Ref: StatQuest: Linear Models - Josh Starmer)

Sum of Squared Errors for Benchmark Line

$$(b - y_1)^2 + (b - y_2)^2 + (b - y_3)^2 + (b - y_4)^2 + (b - y_5)^2 + (b - y_6)^2 + (b - y_7)^2 + (b - y_8)^2 + (b - y_9)^2$$



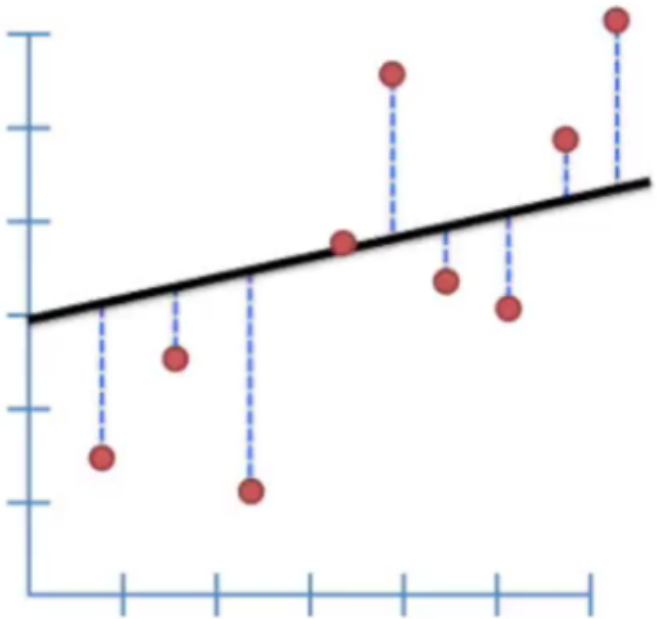
= 24.62

This is our measure of how well this line fits the data.

It's called the "sum of squared residuals, because the residuals are the differences between the real data and the line, and we are summing the square of these values.

(Ref: StatQuest: Linear Models - Josh Starmer)

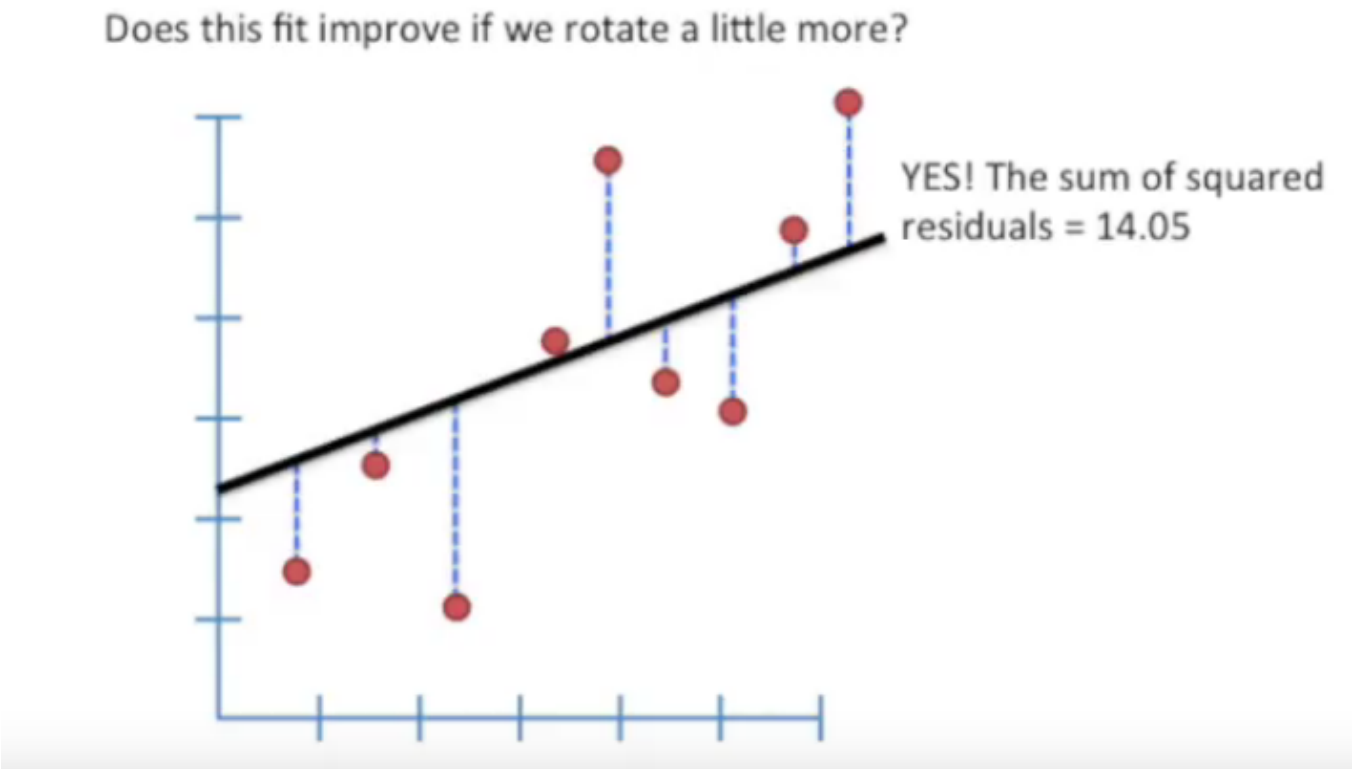
Sum of Squared Errors for Rotated Line



Its better but is this best?
(Ref: StatQuest: Linear Models - Josh Starmer)

Sum of Squared Errors for Rotated Line

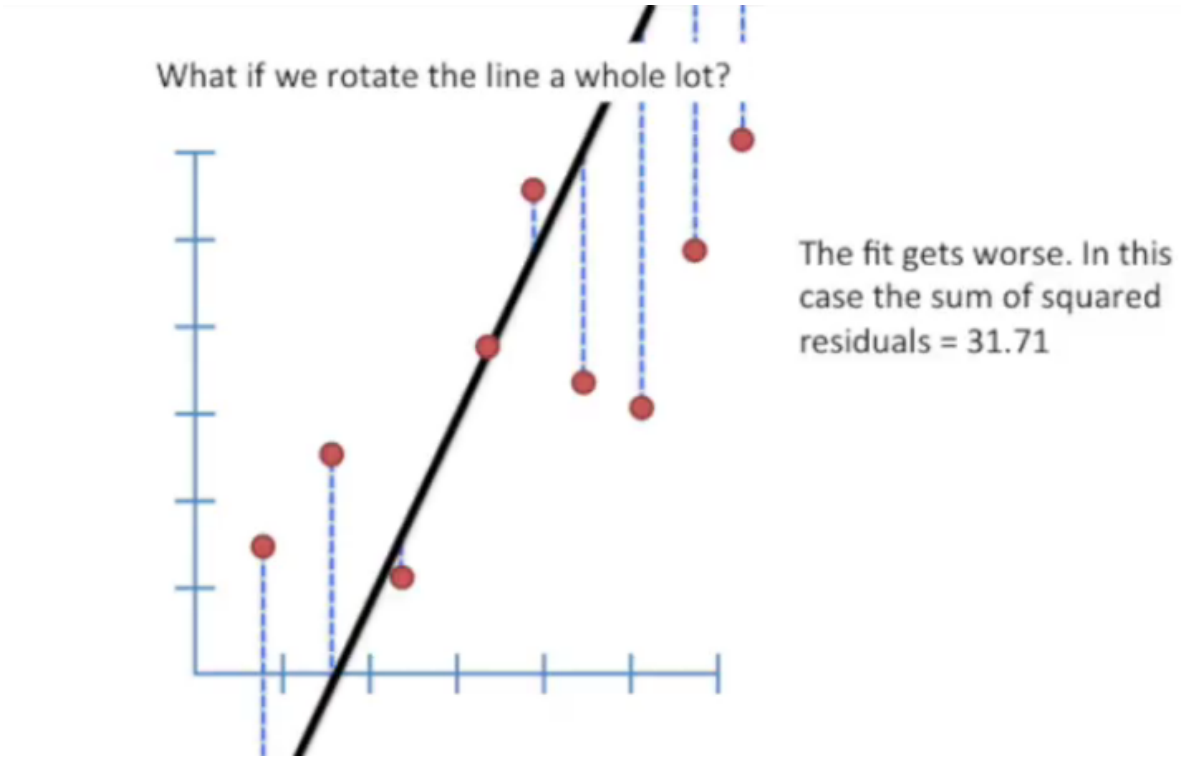
Rotated a bit more. Better?



Its better but is this best?
(Ref: StatQuest: Linear Models - Josh Starmer)

Sum of Squared Errors for Rotated Line

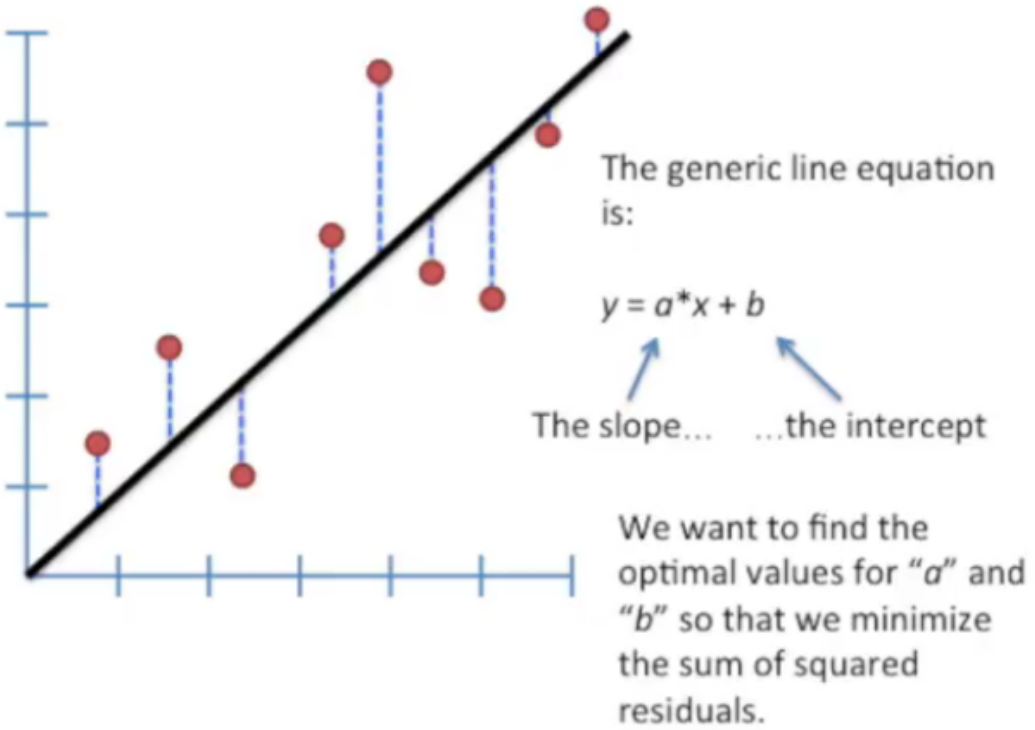
Why not rotate a whole lot!!



No, we lost the BEST point somewhere in between. Whats that "optimized" point?
(Ref: StatQuest: Linear Models - Josh Starmer)

Optimization

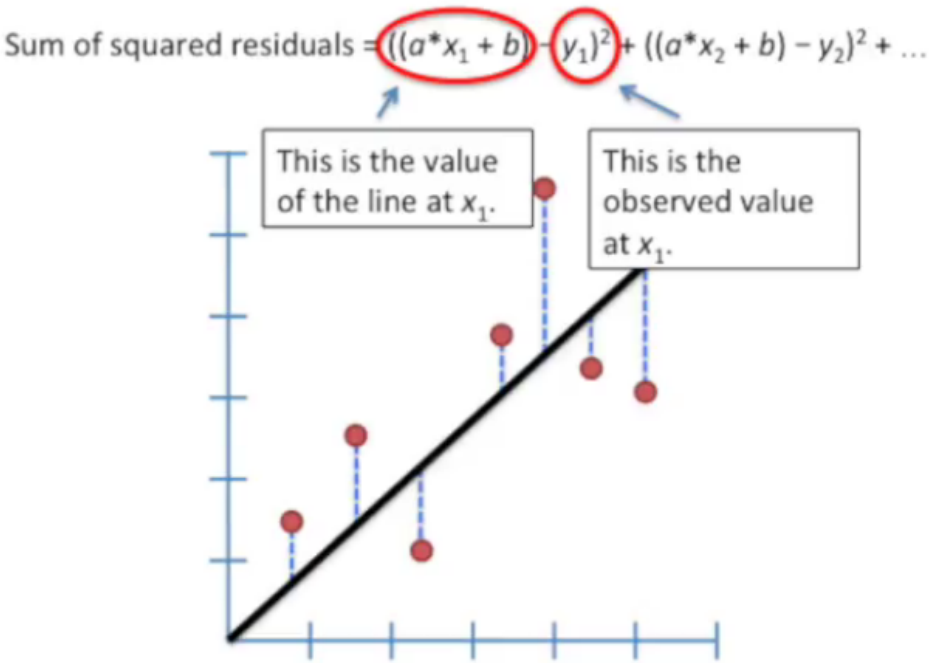
To find the sweet spot ie the optimized point, lets put some formulation based on generic line equation.



(Ref: StatQuest: Linear Models - Josh Starmer)

Optimization

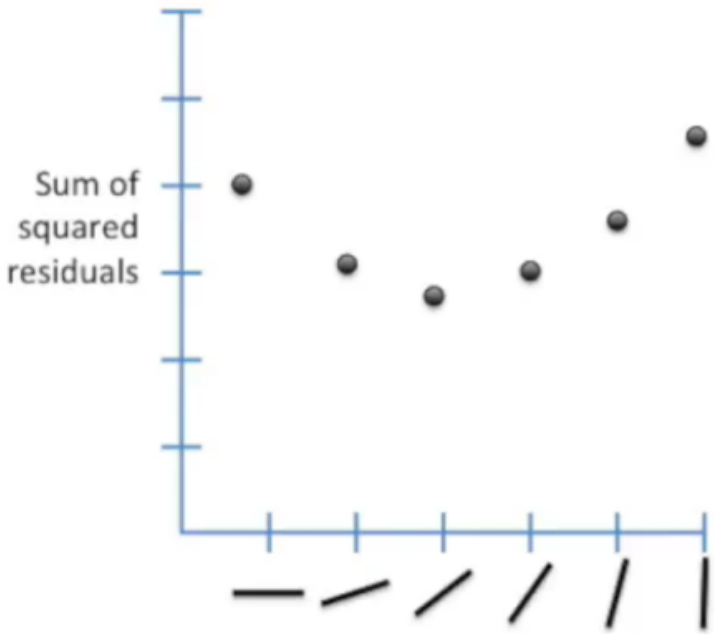
Mathematically . . .



Since we want the line that will give us the smallest sum of the squares of errors (SSE: Sum of Squared Errors), this method for finding the best values for “a” and “b” is called as “Least Squares”.
(Ref: StatQuest: Linear Models - Josh Starmer)

Optimization

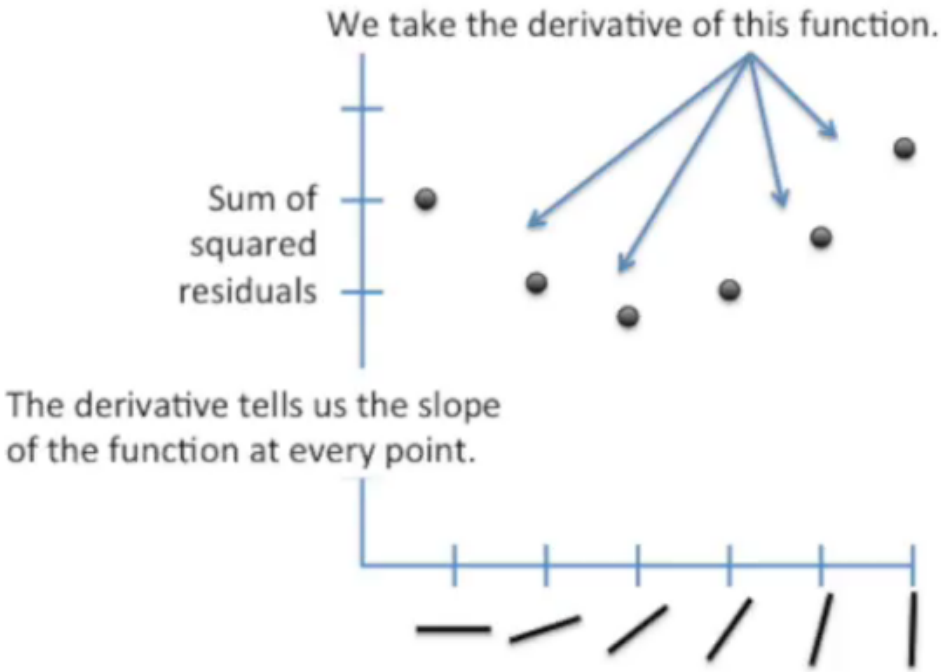
- If we plotted SSE for each rotated line the plot looks like.
- We can see that SSE was high for the horizontal line, then came down as we started rotating the line,
- But started rising again, if we passed the sweet spot.



(Ref: StatQuest: Linear Models - Josh Starmer)

Optimization

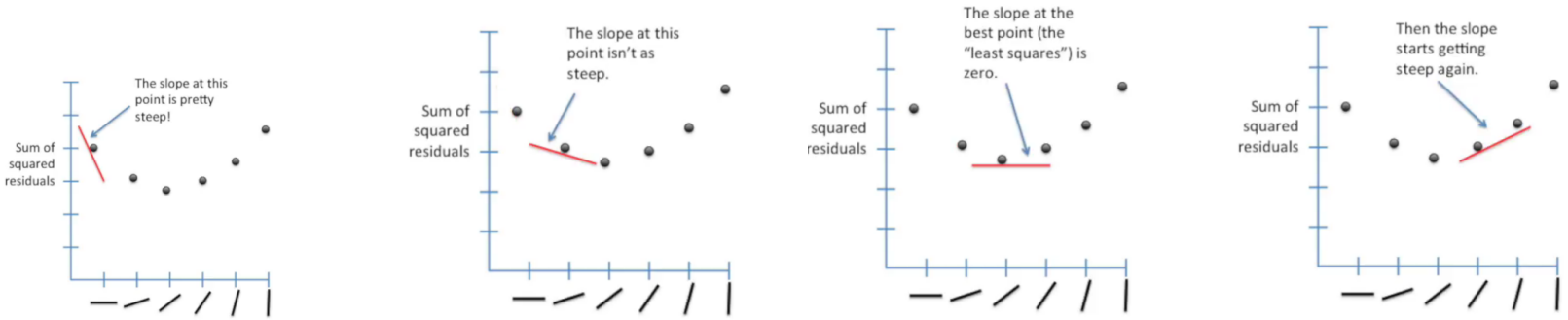
- We need to get to the bottom point.
- We can start at any (random) point, and SLIDE down to bottom.
- SLIDING means going along slope.
- Slope is given by Derivative of the (SSE) function/curve, at the point



(Ref: StatQuest: Linear Models - Josh Starmer)

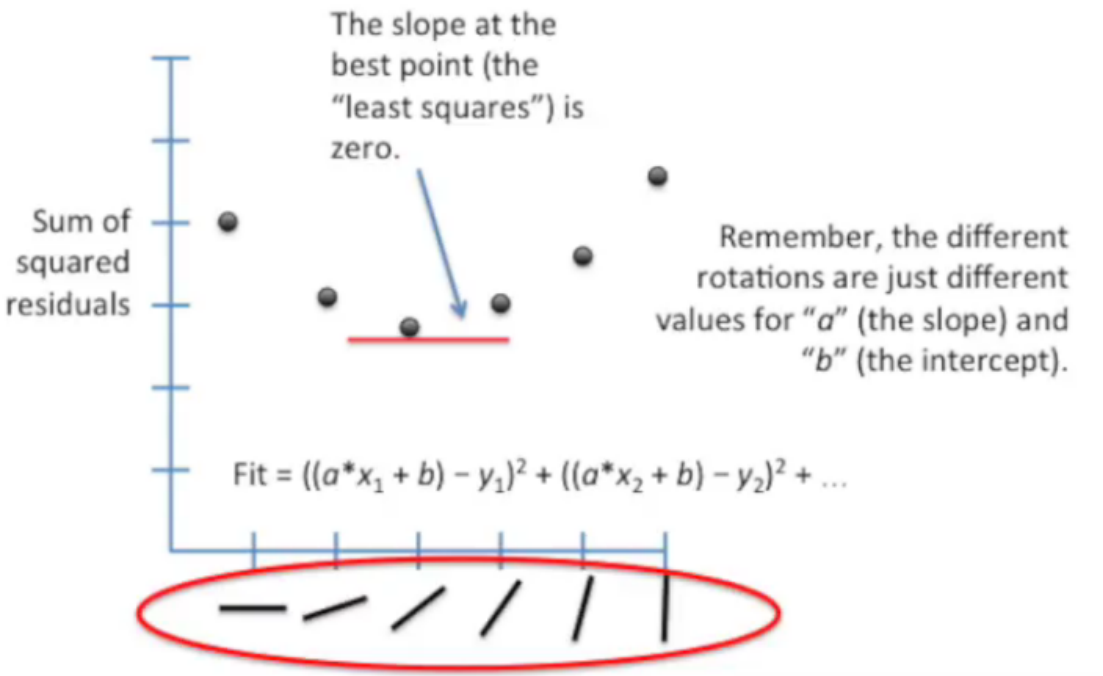
Optimization

- Slope starts decreasing from end, faster,then slower,
- Reaches 0, horizontal, and the starts increasing again.
- Need to find the “0” slope point.



These different slopes means different “a” parameters (Ref: StatQuest: Linear Models - Josh Starmer)

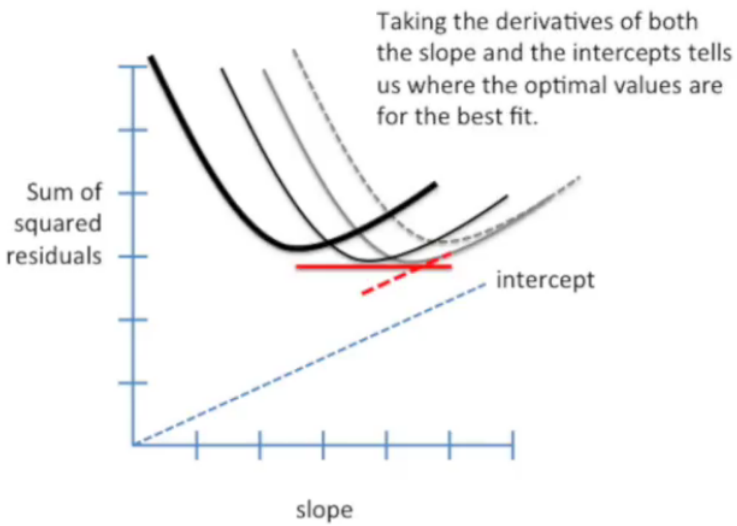
Optimization



(Ref: StatQuest: Linear Models - Josh Starmer)

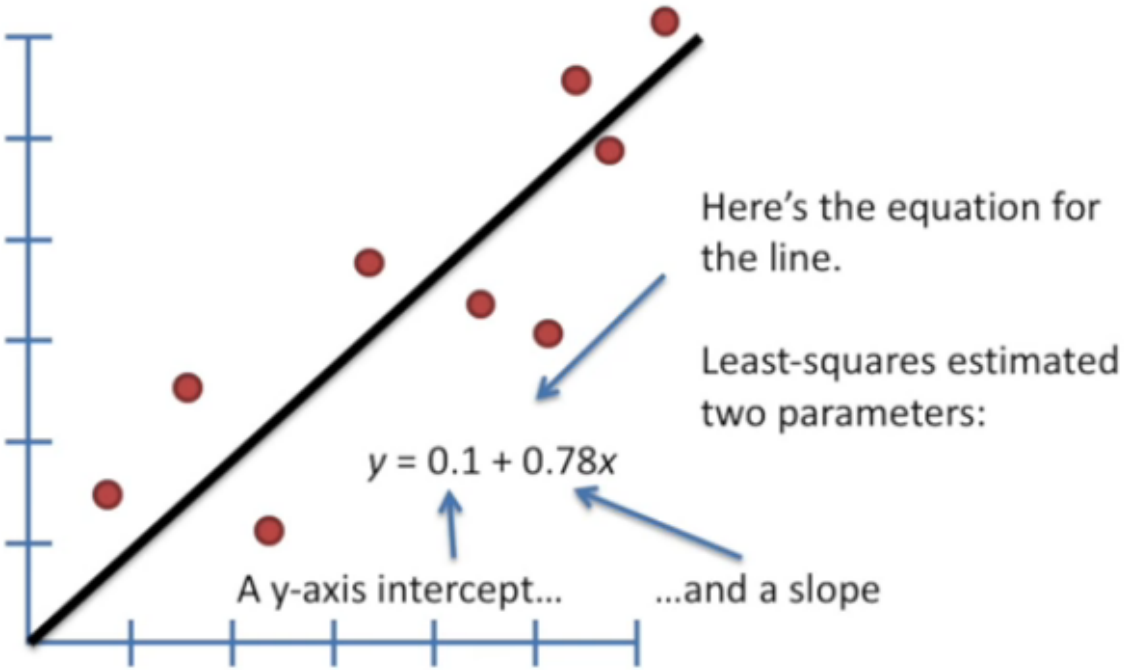
Optimization

- For “a” and “b” both to be optimized, we will need 3D picture, as the Loss function would then take Z axis
- It will be a Surface.
- Derivative will be Partial, but the objective is the same, Bottom-most point. ie Minimum Loss function value



The ’“a” and “b” coordinates corresponding to the bottom-most point give most minimum Loss. So it represents the “best” line.
(Ref: StatQuest: Linear Models - Josh Starmer)

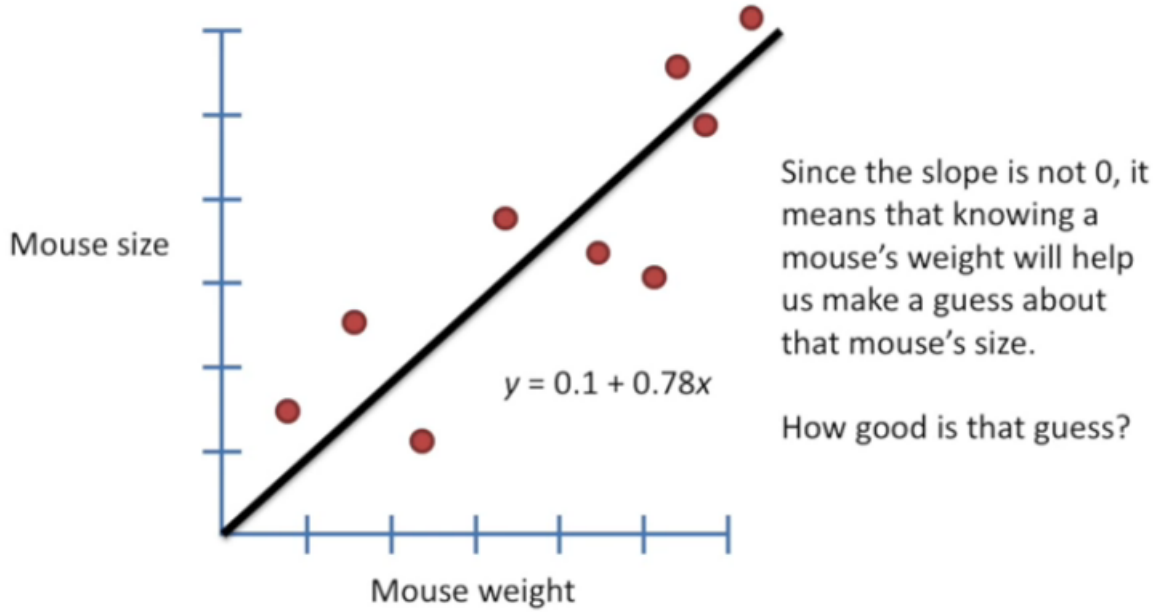
Results



(Ref: StatQuest: Linear Models - Josh Starmer)

Example: Mouse Size

- We have data points having mouse size and mouse weight
- We wish to find mouse size given mouse weight.



we use R^2 to find "good-ness" of fit (of line)
(Ref: StatQuest: Linear Models - Josh Starmer)

Linear regression For Complex Problems

- In real life, problems are far more complex that just one x and one y.
- There could be many x's ie many inputs which decide the output
- General formulation : Predicting quantitative response y based on predictor variables x. Assumes linear relationship between x and y

$$y_i = x_{1,i}\beta_1 + x_{2,i}\beta_2 + \cdots + x_{p,1}\beta_p + \epsilon_i$$

- Objective is to find the parameters β_j that minimize the squared residuals
- There is no slope formulas for these βs. So, we will solve machine learning way.

Linear Regression - Advertising Budget Example

Example: Advertising Dataset

- Toy Dataset: Advertising, Advertising.csv
- Sales totals for a product in 200 different markets. Advertising budget in each market, broken down into TV, radio, newspaper

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

(Data: http://www-bcf.usc.edu/g̃areth/ISL/data.html)

Problem

- Goal: What marketing plan for next year will result in high product sales?
- Questions (Hypothesis):
  - Is there a relationship between advertising budgets and sales?
  - How strong is the relationship between advertising budgets and sales?
  - Which media contribute to the sales the most?
  - Is the relationship linear?

EDA

Correlation Matrix

	TV	Radio	Newspaper	Sales
TV	1.000000	0.054809	0.056648	0.782224
Radio	0.054809	1.000000	0.354104	0.576223
Newspaper	0.056648	0.354104	1.000000	0.228299
Sales	0.782224	0.576223	0.228299	1.000000

- Correlation between radio and newspaper is 0.35
- Barely any correlation (or "not correlated") for TV/radio and TV/newspaper

EDA

	TV	Radio	Newspaper	Sales
TV	1.000000	0.054809	0.056648	0.782224
Radio	0.054809	1.000000	0.354104	0.576223
Newspaper	0.056648	0.354104	1.000000	0.228299
Sales	0.782224	0.576223	0.228299	1.000000

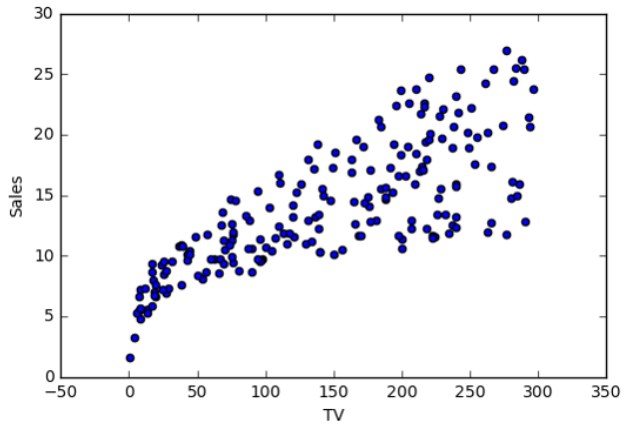
- Reveals tendency to spend more on Newspaper advertising in markets where more is spent on Radio advertising.
- Sales higher in markets where more is spent on Radio, but more also tends to be spent on Newspaper.
- In Simple LM: Newspaper "gets credit" for effect of Radio on Sales.

Advertising Dataset

- What are some ways we can regress sales onto adverting using Simple Linear Regression?
- One model:

$$\begin{aligned} \textit{sales} &\approx \beta_0 + \beta_1 \times \textit{TV} \\ y &\approx \beta_0 + \beta_1 \times x \end{aligned}$$

- Scatter plot visualization for TV and Sales.

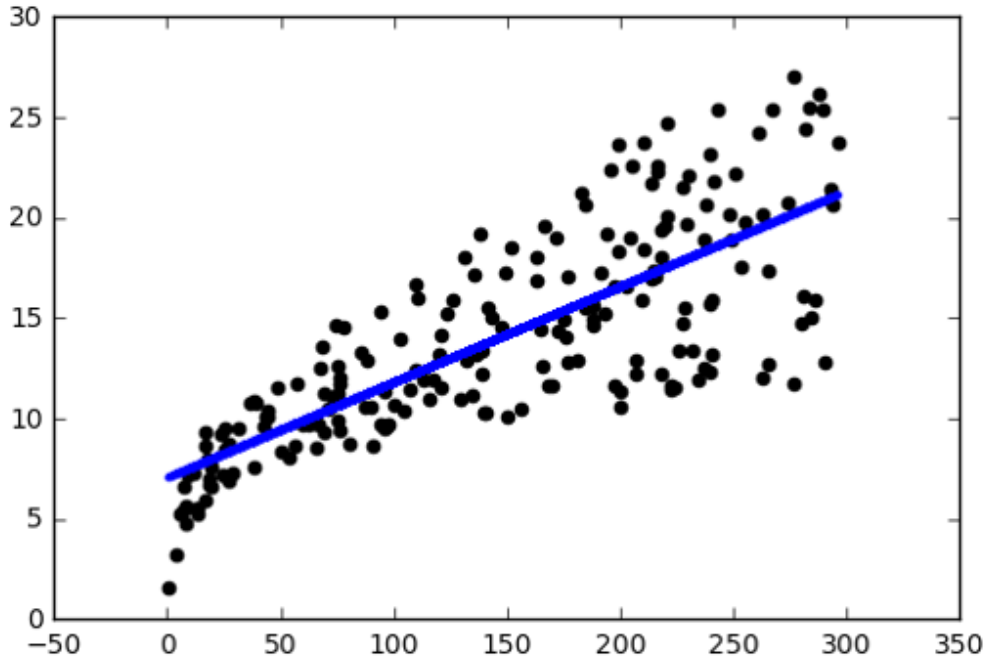


Advertising Dataset

- Simple Linear Model in Python (using pandas and scikit): Predictor: x, Response: y
- $Sales = 7.03259 + 0.04754 \times TV$

Advertising Dataset

Scatter plot visualization for TV and Sales with Linear Model.



If we have 3 such models, can we "somehow" compose them (say, averaging) to get the combined model?

Multiple Linear Regression

- In practice, often have more than one predictor
  - $sales \approx \beta_{0,1} + \beta_{1,1} \times TV$
  - $sales \approx \beta_{0,2} + \beta_{1,2} \times Newspaper$
  - $sales \approx \beta_{0,3} + \beta_{1,3} \times radio$
- Run three separate simple linear regressions
- For p predictor variables: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$
- Since error ϵ has mean zero, we usually omit it.
- A one-unit change in any predictor variable x_i will change the expected mean response by β_j units.
  $sales = \beta_0 + \beta_1 \times TV + \beta_2 \times radio + \beta_3 \times newspaper$

Estimating the Parameters $\beta_0, \beta_1, \ldots$

- Advertising Dataset $\textit{sales} = \beta_0 + \beta_1 \times \textit{TV} + \beta_2 \times \textit{radio} + \beta_3 \times \textit{newspaper}$
- After solving
  $\textit{sales} = 2.93 + 0.0457 \times \textit{TV} + 0.188 \times \textit{radio} + -0.0010 \times \textit{newspaper}$

Coefficients

- Many Simple Linear Regression
  - *TVModel* : [[0.04753664]][7.03259355]
  - *RadioModel* : [[0.20249578]][9.3116381]
  - *NewspaperModel* : [[0.0546931]][12.35140707]
- Single Multiple Linear Regression
  - *Coefficients* : $[[0.04576465, 0.18853002, -0.00103749]]$
  - *Intercept* : [2.93888937]

Answering questions using Regression Model

Goal: What marketing plan for next year will result in high product sales?

- Question: Is there a relationship between advertising budget and sales?
- Answer: Yes, hypothesis testing shows that we can reject the null hypothesis that $B_{TV} = B_{RADIO} = B_{NEWSPAPER} = 0$

In conclusion

- Pros of Linear Regression Model:
  - Provides nice interpret-able results
  - Works well on many real-world problems
- Cons of Linear Regression Model:
  - Assumes linear relationship between response and predictors: Change in the response Y due to a one-unit change in Xi is constant
  - Assumes additive relationship. Effect of changes in a predictor Xi on response Y is independent of the values of the other predictors

Linear Regression with Scikit-Learn

Linear Regression

```
import pandas
from sklearn import model_selection
from sklearn.linear_model import LinearRegression
url =
     "https://raw.githubusercontent.com/jbrownlee/Datasets/master/housing.data"
names = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
     'TAX', 'PTRATIO', 'B', 'LSTAT', 'MEDV']
dataframe = pandas.read_csv(url, delim_whitespace=True, names=names)
array = dataframe.values
X = array[:,0:13]
Y = array[:,13]
seed = 7
kfold = model_selection.KFold(n_splits=10, random_state=seed)
model = LinearRegression()
scoring = 'neg_mean_absolute_error'
results = model_selection.cross_val_score(model, X, Y, cv=kfold,
     scoring=scoring)
print("MAE: %.3f (%.3f)") % (results.mean(), results.std())

(Ref: Metrics To Evaluate Machine Learning Algorithms in Python - Jason Brownlee)
MAE: -4.005 (2.084)
```

Introduction

Answering yes/no questions

Often we have to resolve questions with binary or yes/no outcomes.

- Does a patient have cancer?
- Will a team win the next game?
- Will the customer buy my product?
- Will I get the loan?

Continuous vs Categorical Variables

Types of variables:

- Continuous: age, income, height : use numerical (float) value
- Categorical (discrete): gender, city, ethnicity : use enums (ints to an extent)

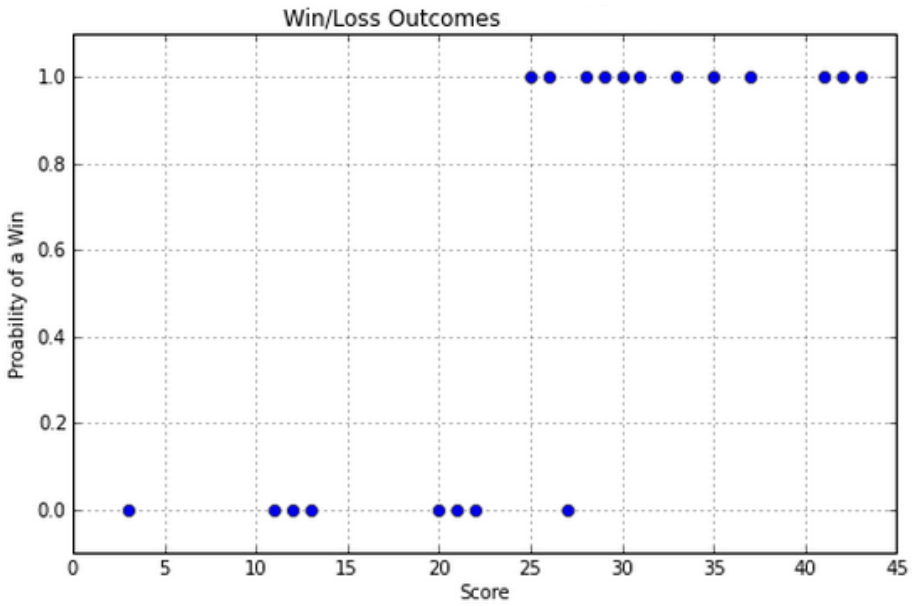
Example: Win/Lose based on Scores

Start by plotting variables that decide the outcome on x axis and the outcome on the y axis.

- X-axis is the number of points scored by a team
- Y-axis is whether they lost or won the game (0 or 1)
- General idea: more the score, more chances of winning, and vice versa.

Any guesses on how the plot would look like?

Plot



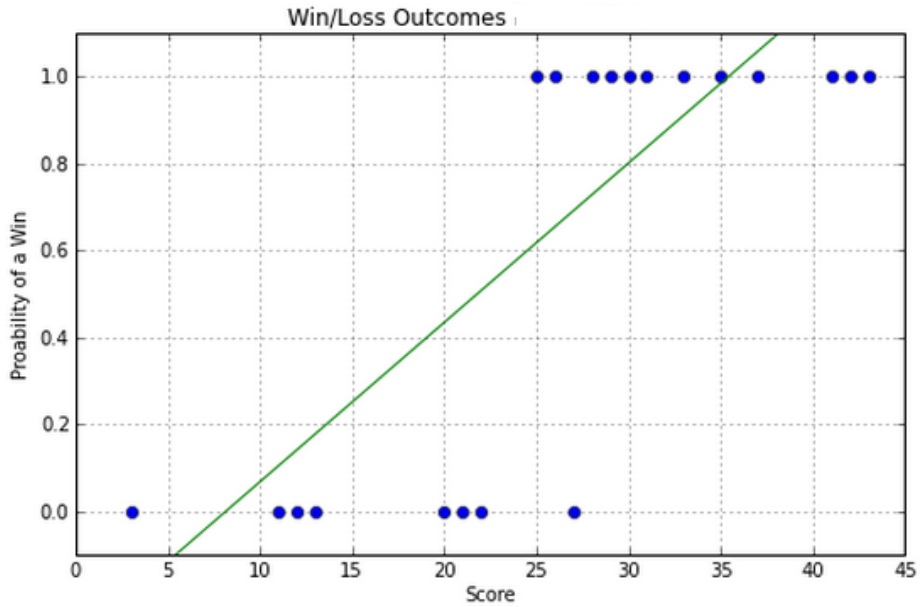
So, how do we predict whether we have a win or a loss if we are given a score?

Game score prediction

- Can you fit a line?
- Sure, you can, but is represent the data?
- What would be a good representation/model?

Game score prediction

Take a look at this plot of a "best fit" line over the points.

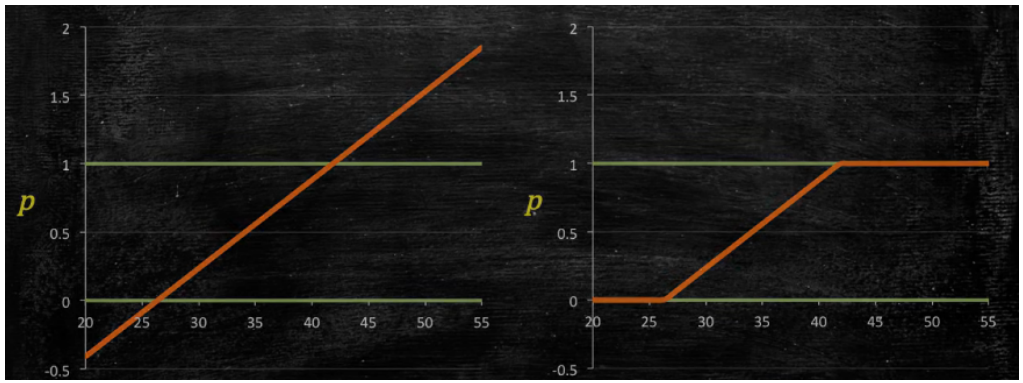


Why not Linear Regression, here?

- Probability of winning is bound, from 0 to 1.
- Scores can range from 0 to 50, say.
- Now we need a function (or a model) that, given a new score, say, 30. it should compute value between 0 to 1. From the graph, it comes out to be 0.8. Its close to 1, so we can predict that the team will win.
- Similarly, for score 10, probability is 0.08, very low, so we can predict that the team will lose.
- But what for scores, 2, or 45. The probabilities are coming out of the range of 0 to 1. Bad!!!

Why not Linear Regression, here?

- Well, we can say, value calculated above 1 is 1, and similarly any value calculated below 0 is considered to be 0. Some sort of Capping.
- Then the function would look like mirrored "Z". It has kinks. Too specific!!, not good.
- So, clearly linear regression is not a good model.
- Can we find a better function?



What do we need here?

- We need some magic function to represent this requirement.
- Input: any number
- Output: either Boolean or even probabilities between 0 to 1 would do. Using threshold we will convert them to Booleans.

Logistic Regression

- For example, consider a logistic regression model for spam detection.
- If the model infers a value of 0.932 on a particular email message, it implies a 93.2% probability that the email message is spam.
- More precisely, it means that in the limit of infinite training examples, the set of examples for which the model predicts 0.932 will actually be spam 93.2% of the time and the remaining 6.8% will not.
- Choice of threshold is an important choice, and can be tuned

Logistic Regression

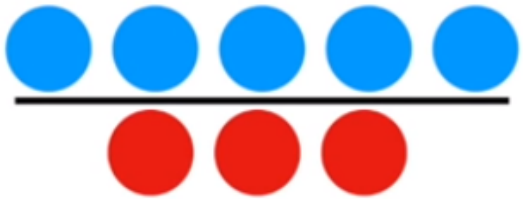
- The name of the Magic function is *logit*.
- The theory is based on a term called "Odds".
- Lets look at them in details, next.

Background Concepts

Odds vs Probabilities

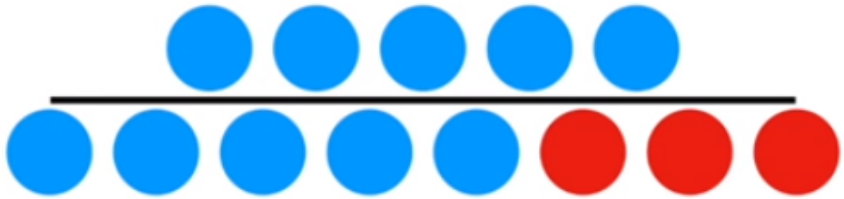
The odds are the ratio of something happening (i.e. my team winning)...

...to something not happening (i.e. my team not winning).



Probability is the ratio of something happening (i.e. my team winning)...

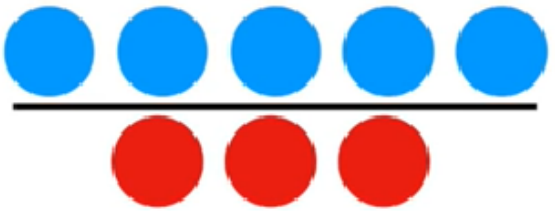
...to *everything* that could happen (i.e. my team winning *and* losing).



(Ref: StatQuest: Odds and Log(Odds), Clearly Explained!!! - Josh Starmer)

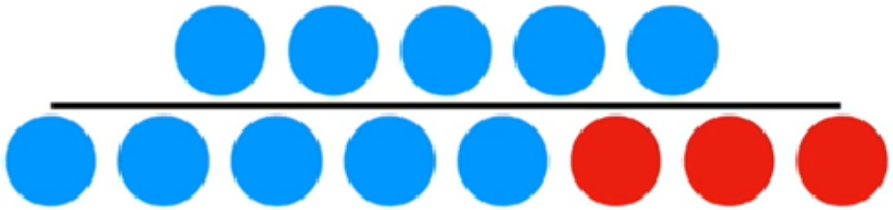
Odds vs Probabilities

In the previous example, the odds in favor of my team **winning** the game are 5 to 3...



$$= \frac{5}{3} = 1.7$$

...however, the probability of my team **winning** is the number of games they win (5) divided by the total number of games they play (8)...



$$= \frac{5}{8} = 0.625$$

...are different from the probability of my team winning.

(Ref: StatQuest: Odds and Log(Odds), Clearly Explained!!! - Josh Starmer)

Probability Basics

$$probability = \frac{one\_outcome}{all\_outcomes} \tag{1}$$

$$odds = \frac{one\_outcome}{all\_other\_outcomes} \tag{2}$$

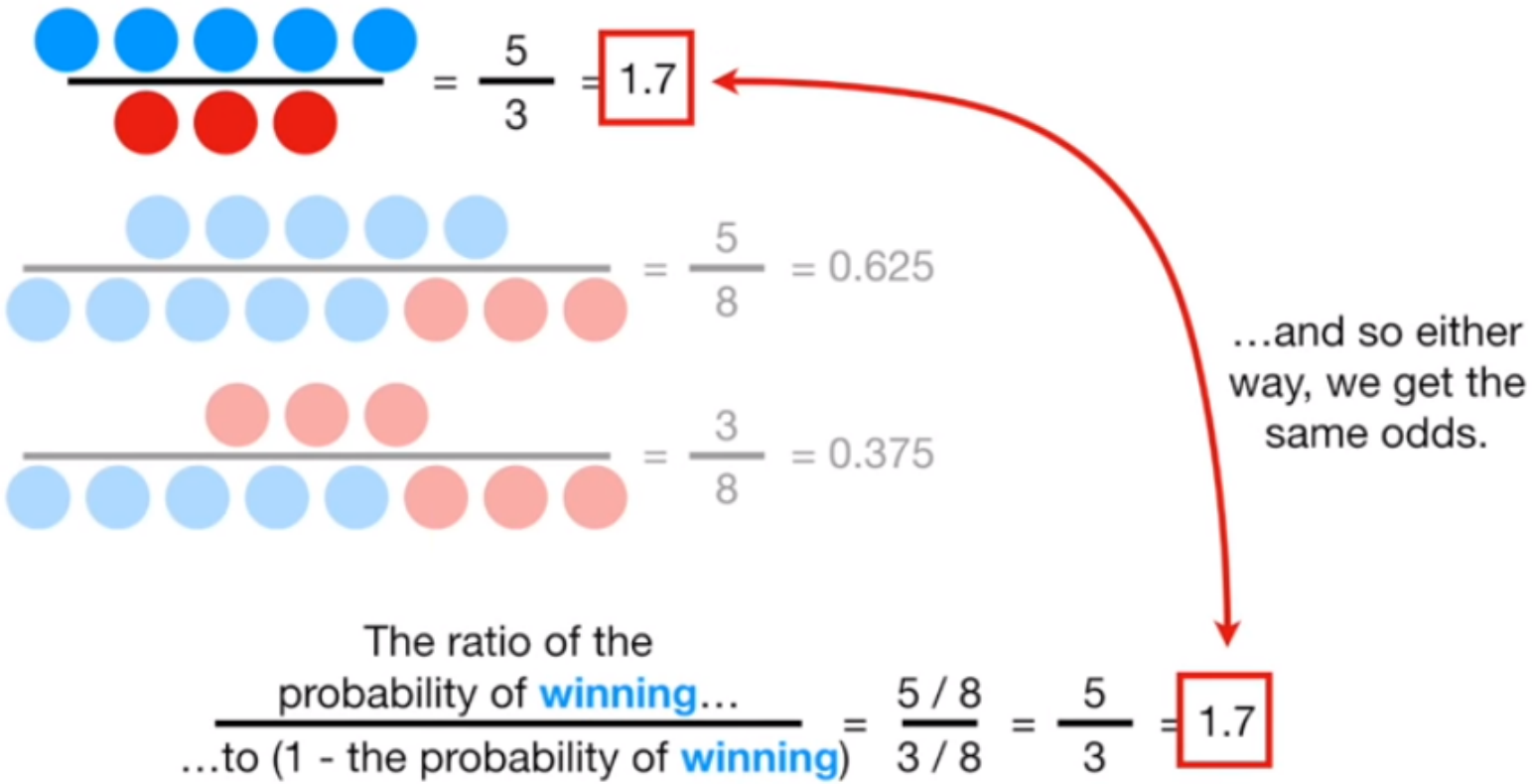
- Dice roll of 1: probability $= 1/6$, odds $= 1/5$
- Even dice roll: probability $= 3/6$, odds $= 3/3 = 1$
- Dice roll less than 5: probability $= 4/6$, odds $= 4/2 = 2$

$$odds = \frac{probability}{1 - probability} \tag{3}$$

$$probability = \frac{odds}{1 + odds} \tag{4}$$

Odds vs Probabilities

Lets see how odds can be calculated from Probabilities.

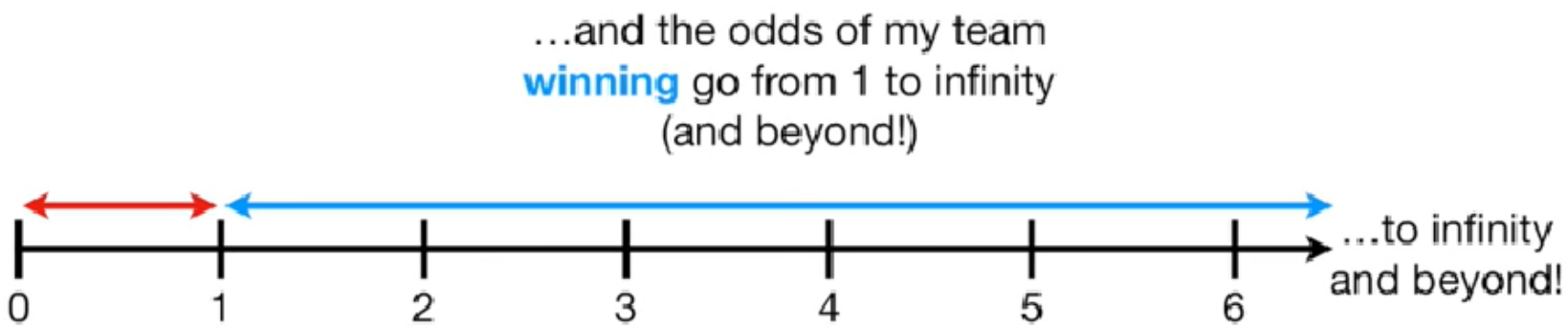


So, $odds = \frac{p}{1-p}$ (Ref: StatQuest: Odds and Log(Odds), Clearly Explained!!! - Josh Starmer)

Odds vs Probabilities

Lets take log of odds. But, why?

- If my team was worse, the odds of winning would be say $\frac{1}{8} = 0.125$
- If my team was terrible, the odds of winning would be say $\frac{1}{16} = 0.062$
- If my team was worst, the odds of winning would be say $\frac{1}{32} = 0.031$
- Worser my team gets the odds get closer to 0. The range is 0 to 1
- Similarly, for winning, the odds start at 1 and just go up and up, reaching infinity.

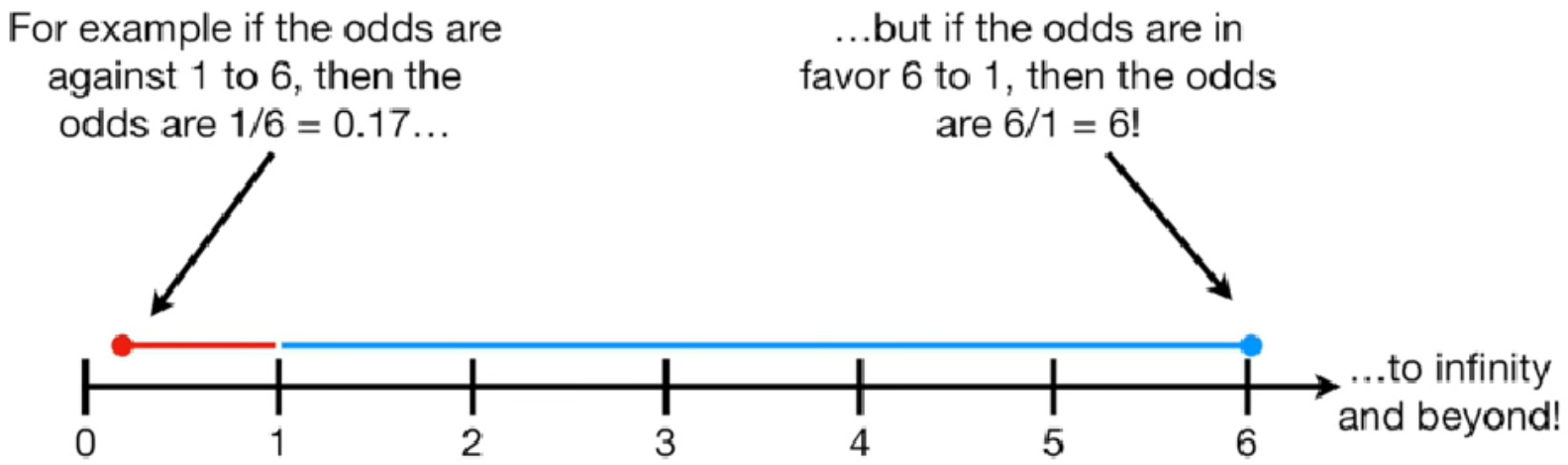


This is asymmetrical.

(Ref: StatQuest: Odds and Log(Odds), Clearly Explained!!! - Josh Starmer)

Odds vs Probabilities

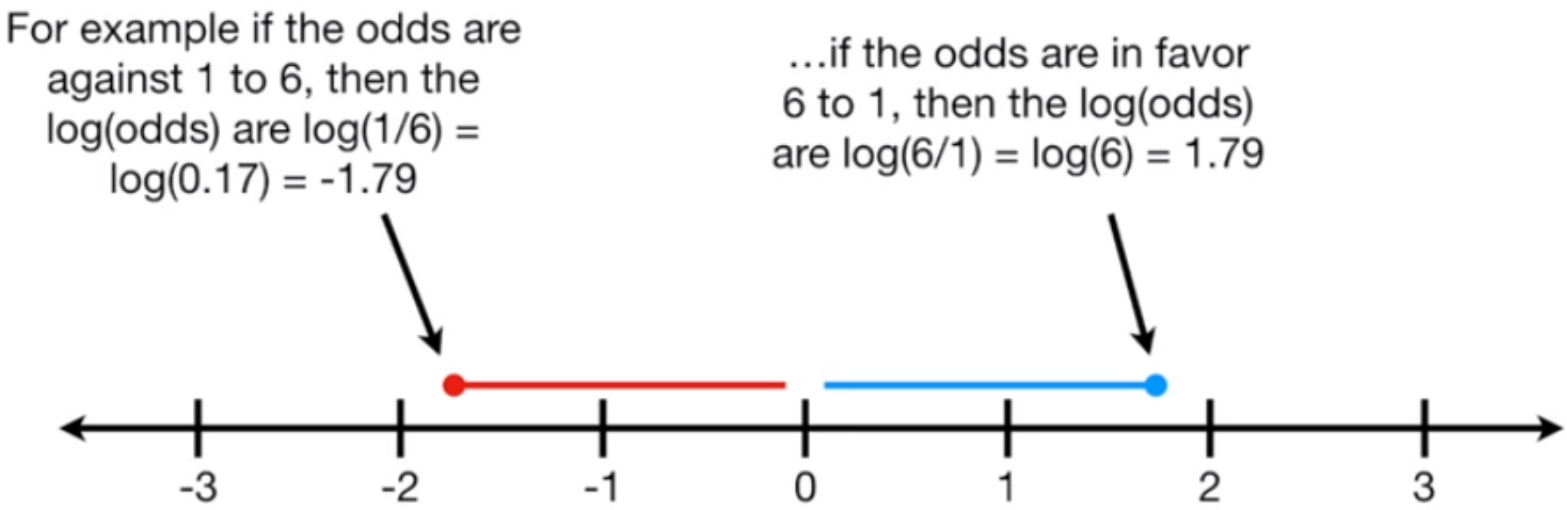
- Asymmetry makes it difficult to compare odds of winning and losing.
- 1 out of 6, chance on wither side, gives hugely different odds.



(Ref: StatQuest: Odds and Log(Odds), Clearly Explained!!! - Josh Starmer)

Odds vs Probabilities

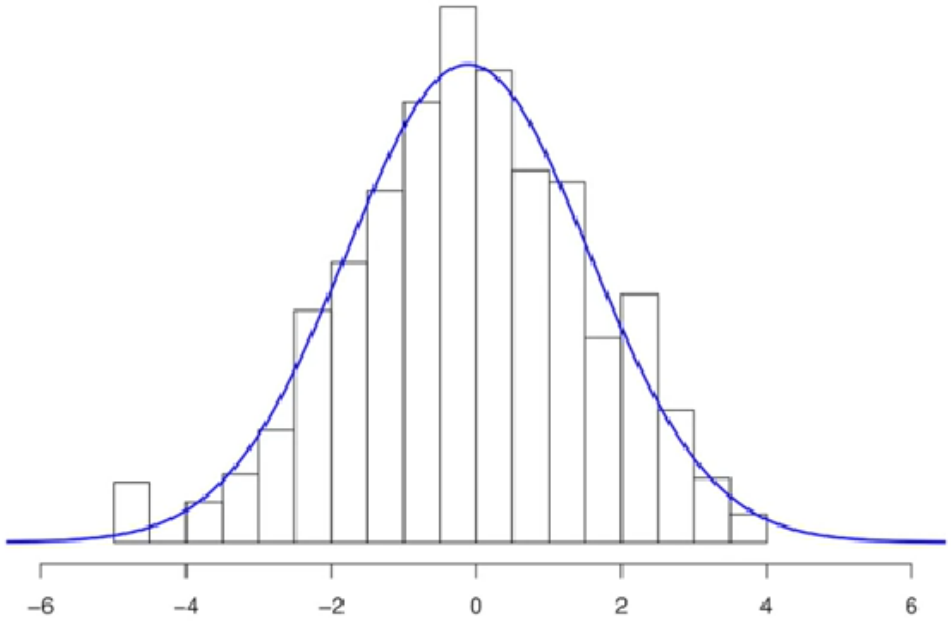
Taking log of odds, solves this problem by making everything symmetrical.



Using log, distance from origin for 6, is same!! (Ref: StatQuest: Odds and Log(Odds), Clearly Explained!!! - Josh Starmer)

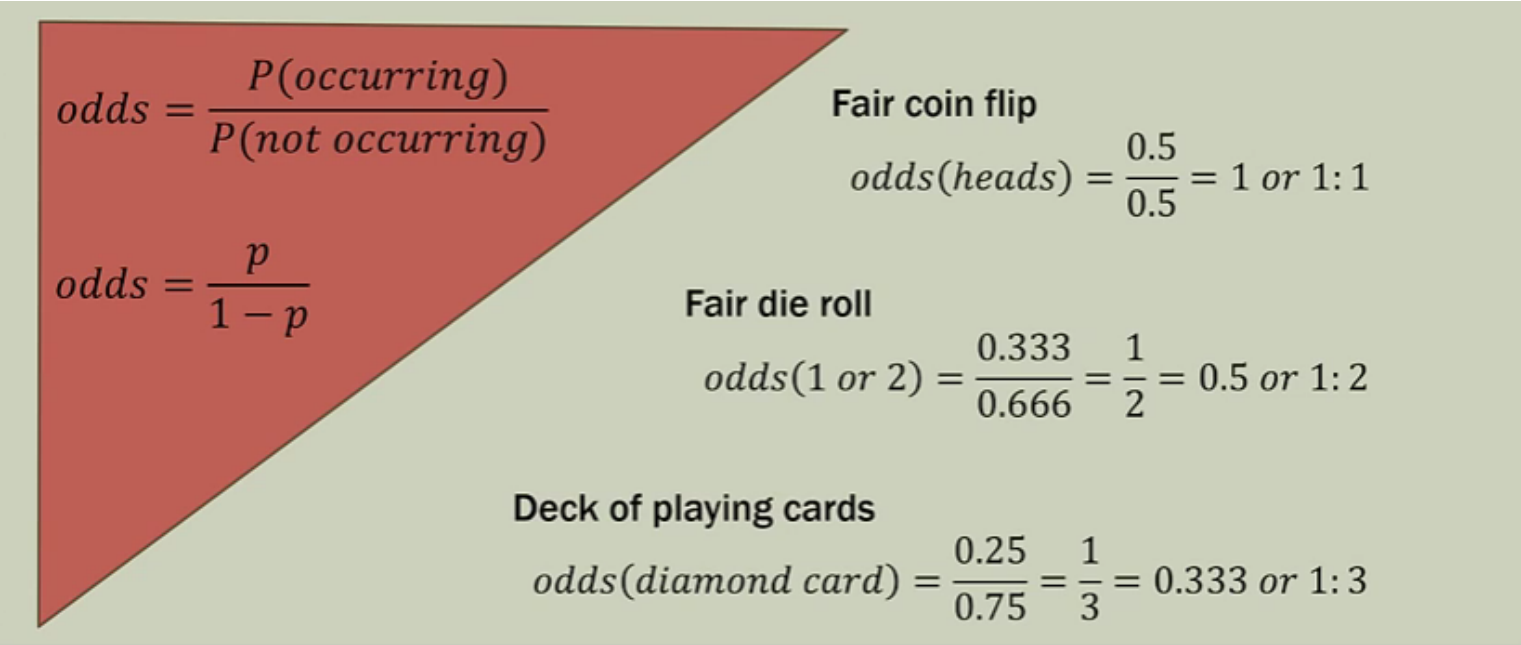
Whats the big deal with this log of odds?

If you pick, random numbers that add up to, say 100 (meaning 20—80, 35—65, this is like winning — losing percentage) and use them to calculate the log(odds) and draw histogram. The histogram is in the shape of normal distribution. The coveted shape of the input data.



(Ref: StatQuest: Odds and Log(Odds), Clearly Explained!!! - Josh Starmer)

So, What are Odds? (Summary)



- Odds: Ratio of something happening to that thing not happening.
- $odds = \frac{p}{(1-p)}$, where, p stands for probability of the positive (the one we want to predict, not necessarily a "good" one) event.

(Reference: Simple Linear Regression: Step-By-Step - Dan Wellisch)

Logit

- p being probability can be only between 0 to 1.
- Taking log of odds, gives the "logit" function $logit(p) = log_e \frac{p}{(1-p)}$
- Whats the range of logit values?

(Reference: Simple Linear Regression: Step-By-Step - Dan Wellisch)

Range of Logit

- Put p $= 0.5$, $logit(0.5) = log\frac{0.5}{(0.5)} = 0$
- Put p $= 0.1$, $logit(0.1) = log\frac{0.1}{(0.9)} = -0.95$
- Put p $= 0.01$, $logit(0.01) = log\frac{0.01}{(0.99)} = -1.99$
- Put p $= 0.9$, $logit(0.9) = log\frac{0.9}{(0.1)} = 0.95$
- Put p $= 0.99$, $logit(0.99) = log\frac{0.99}{(0.01)} = 1.99$
- Put p $= 0.999$, $logit(0.999) = log\frac{0.999}{(0.001)} = 2.99$

So, the logit function takes input values in the range 0 to 1 and transforms them to values over the entire real number range.

Reverse

We want something inverse ...

- $logit(p) = log_e \frac{p}{(1-p)}, p = 0 \rightarrow 1$
- We are interested in reverse
- $\phi(z) = logit^{-1}(z) = \frac{1}{1+e^{-z}}, z = -\infty \rightarrow \infty$
- z is any number. In our case its a linear combination of variables giving any number, and the logit inverse will return the probability
- $logit^{-1}(z)$ is called as Sigmoid function

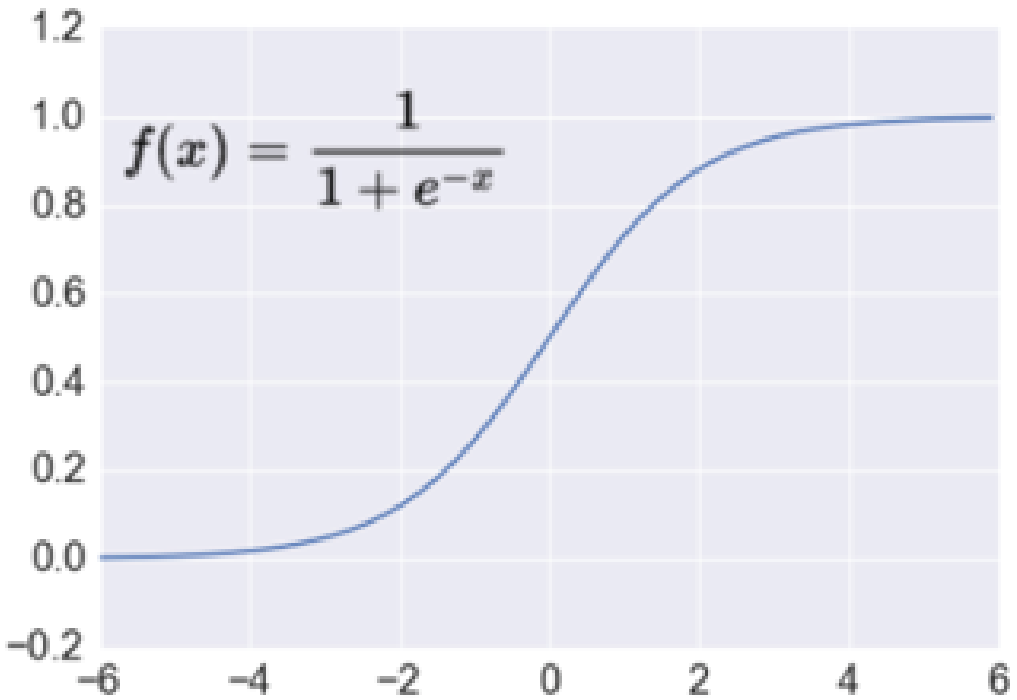
Plotting Sigmoid

```
import matplotlib.pyplot as plt
import numpy as np

def sigmoid(z):
  return 1.0 / (1.0 + np.exp(-z))

z = np.arange(-7, 7, 0.1)
phi_z = sigmoid(z)
plt.plot(z, phi_z)
plt.axvline(0.0, color='k')
plt.axhspan(0.0, 1.0, facecolor='1.0', alpha=1.0, ls='dotted')
plt.axhline(y=0.5, ls='dotted', color='k')
plt.yticks([0.0, 0.5, 1.0])
plt.ylim(-0.1, 1.1)
plt.xlabel('z')
plt.ylabel('$\phi (z)$')
plt.show()
```

Sigmoid



Sigmoid

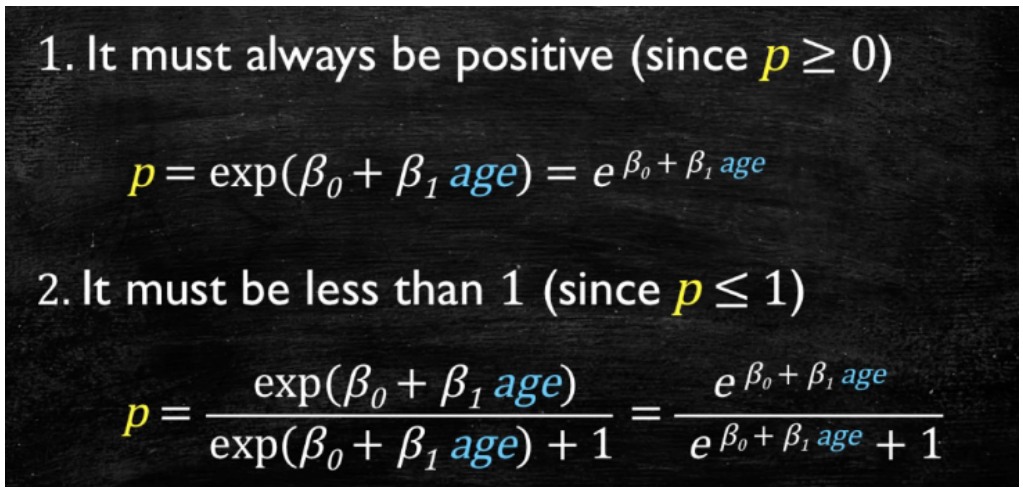
$\phi(z) = logit^{-1}(z) = \frac{1}{1+e^{-z}}$

- If z goes towards infinity then e^{-z} becomes very small , denominator almost 1, thus $\phi(z)$ becomes almost 1.
- If z goes towards minus infinity then e^{-z} becomes very large , denominator almost infinity, thus $\phi(z)$ becomes almost 0.
- Thus, we conclude that this sigmoid function takes real number values as input and transforms them to values in the range [0, 1] with an intercept at $\phi(z) = 0.5$

Logistic Regression Algorithm

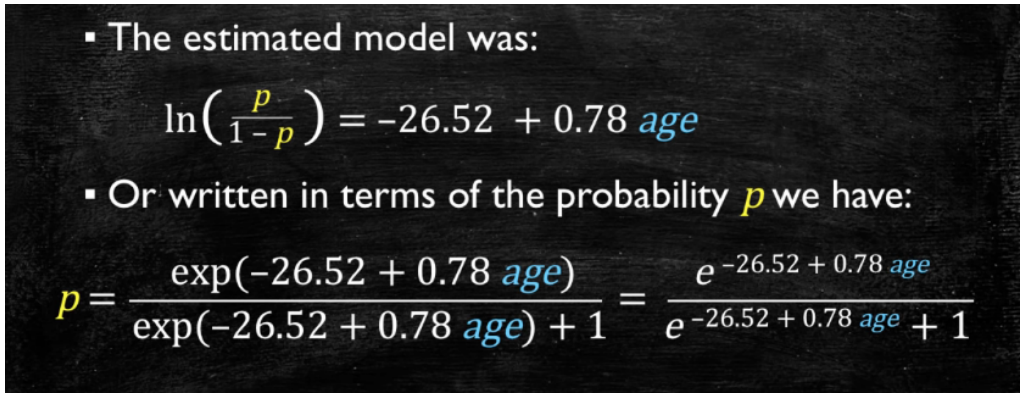
The Magic function

- Example problem, say, that chances of you getting Diabetes is more with age. So, we are finding a function that takes expression $\beta_0 + \beta_1 age$ as input and outputs a probability.
- For probability, 2 conditions are that the value is always positive and less than 1.
- Which functions always output positive value given any (positive or negative) value?
- 'Absolute', 'Square', etc. One more is exponential.
- e^z is always positive whatever the z is. But it can be greater than 1.
- To make it less than 1, lets divide the same term but slightly larger, say, with plus 1. Nice Trick!!



The Magic function

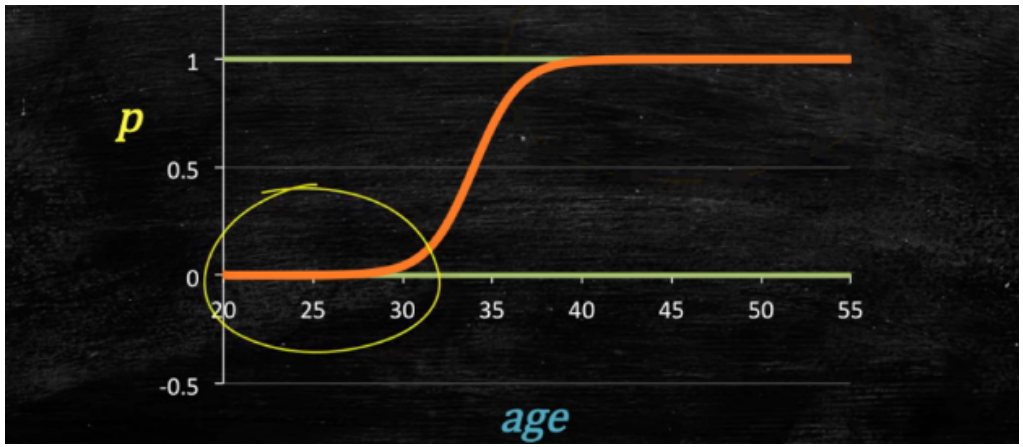
- Rearranging previous equation.
- Lets say $z = \beta_0 + \beta_1 age$. So we have $p = \frac{e^z}{e^z+1}$
- Is it same as $p = \frac{1}{1+e^{-z}}$? Can not believe? Try.
- Arranging p terms and taking log on both sides $ln(\frac{p}{1-p}) = z = \beta_0 + \beta_1 age$
- So even though the probability function is not linear (its exponential), the above arrangement is Linear function of age. Very similar to Linear Regression.



(Ref: Logistic Regression - Interpretation of Coefficients and Forecasting - Data Mining In CAE)

The Magic function

- Plot of Probability vs age looks like a Signmod
- As the age increases, the chances of getting diabetes are almost equal to 1 (Asymptotic).
- As the age lowers, the chances of getting diabetes are almost equal to 0 (Asymptotic).
- $logit(p) = \log(p/(1-p)) = b_0 + b_1 x_1 + \ldots + b_k x_k$
- The $ln(\frac{p}{1-p})$ is the logit function and thus the algorithm is called as Logistic.
- It has Regression expression on the right, so its called Regression.
- Thus this is Logistic Regression. Used for Classification



(Ref: Logistic Regression - Interpretation of Coefficients and Forecasting - Data Mining In CAE)

Intuition ...

- Whats the meaning of Linear expression's coefficients?
- In Linear Regression, it meant slope, or proportionality of increase of y wrt x.
- Here thats not the case.
- Every unit increase in *age*, $\log(p/(1-p))$ increases β_1 units.
- Say, for age as 35, $\beta_0 = -26.52$ and $\beta_1 = 0.78$, the regression comes to be $= -26,52 + 0.78 \times 35 = 0.813$.
- Is it Probability of having Diabetes??
- No. Its logit value.
- $p = \frac{e^z}{e^z+1}$ where $z = 0.813$.
- Result: Probability $p = 0.693$
- Approx 70% chance.
- Exercise: Calculate for age 36.
- Confirm: $z = 1.594$ and Probability comes to be 0.831. Just a marginal increase in age, far more chances of getting disease. But after certain age, say 45, the probability settles towards 1.

Intuition ...

- You can add one more variable to the regression part.
- We add Gender.
- $z = \beta_0 + \beta_1 age + \beta_1 woman$
- Assume that betas are calculated and the equation becomes: $z = -26.47 + 0.79 \times age - 0.56 \times woman$
- One year increase in Woman's age increases risk by 14%, and thats 12% for male.

Variable	Coefficients	35y W	35y M	36y W	36 M
Constant	-26.465300	1	1	1	1
Age	0.787213	35	35	36	36
Woman	-0.557795	1	0	1	0
y* = ln(p/(1-p))		0.529	1.087	1.317	1.874
p = exp(y*)/(exp(y*)+1)		0.629	0.748	0.789	0.867

(Ref: Logistic Regression - Interpretation of Coefficients and Forecasting - Data Mining In CAE)

Optimization ...

- $logit(p) = \log(p/(1-p)) = b_0 + b_1 x_1 + \ldots + b_k x_k$
- Above, p is the probability of presence of the characteristic of interest.
- It chooses parameters that maximize the likelihood of observing the sample values rather than that minimize the sum of squared errors (like in ordinary regression).

Conclusion ...

For better understanding ...

- So, by now it is clear that the Logistic Regression is somewhat a misnomer!
- It is a classification not a regression algorithm problems.
- We can imagine that the Logistic Regression is done in two parts.
- In first, using X features, some intermediate y value is predicted, like Regression.
- This value is then passed via this Analog to Digital Converter like Sigmoid function, resulting into 0 or 1.

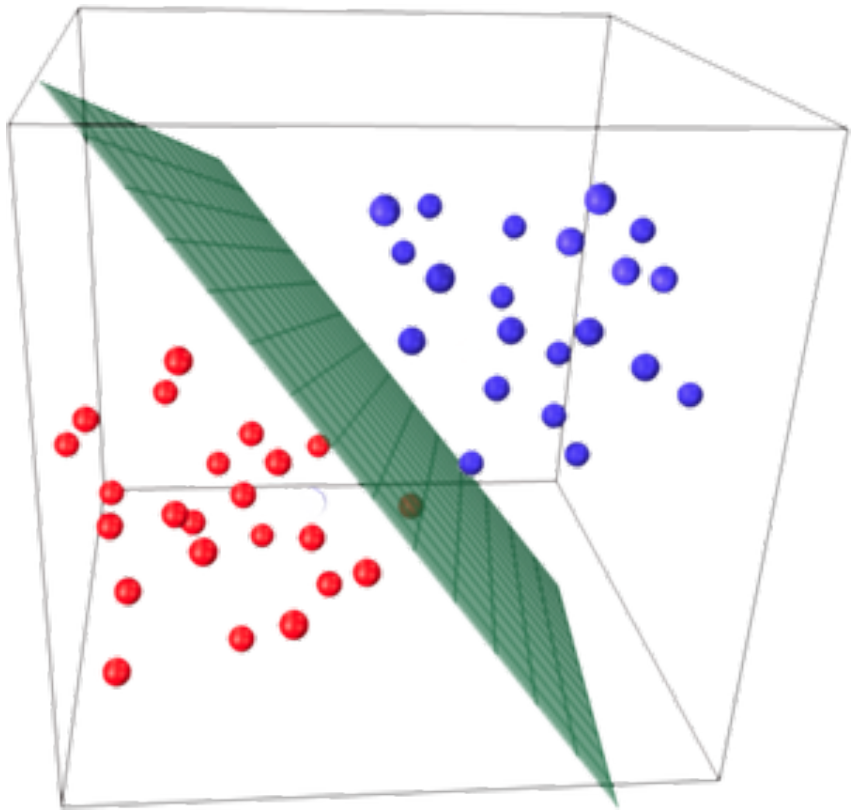
Hyperplane Approach (Advance)

(Ref: Logistic Regression (for dummies) - Sachin Joglekar)

Logistic Regression

- Unlike actual regression, logistic regression does not try to predict the value of a numeric variable given a set of inputs. Instead, the output is a probability that the given input point belongs to a certain class.
- Your input space can be separated into two nice regions, one for each class, by a linear(read: straight) boundary.
- Equation of the separating Hyperplane is given by $W^T x = 0$
- Whereas $W^T x$ gives a distance of a x point from $W^T x = 0$ Hyperplane.

Logistic Regression Visualization



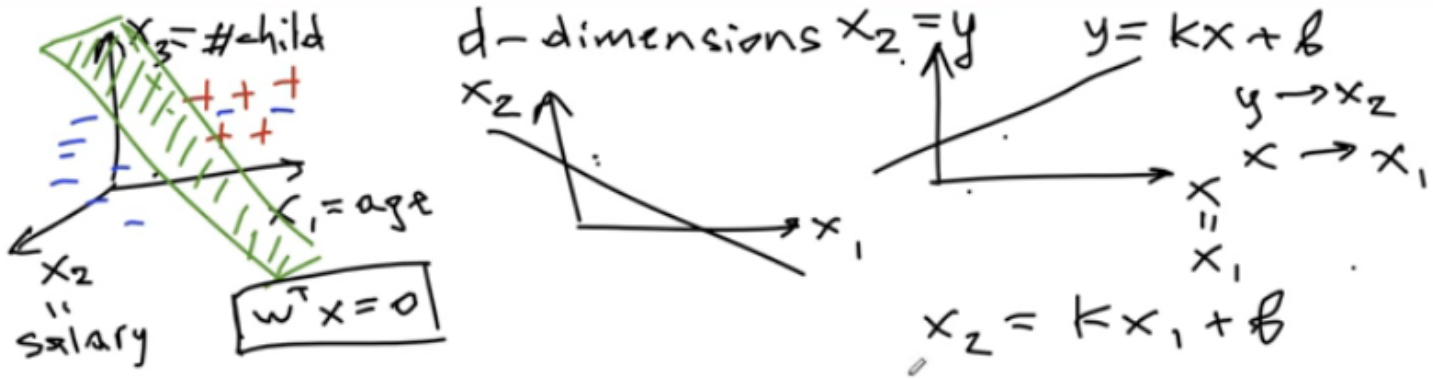
$W^T x = 0$ is a generalization for n dimensions with both w and x as n dimensional vectors and the Hyperplane is the dot product.

(Ref: http://blog.sairahul.com/2014/01/linear-separability.html)

Logistic Regression

Lets look at 2D example, as a special, simpler case:

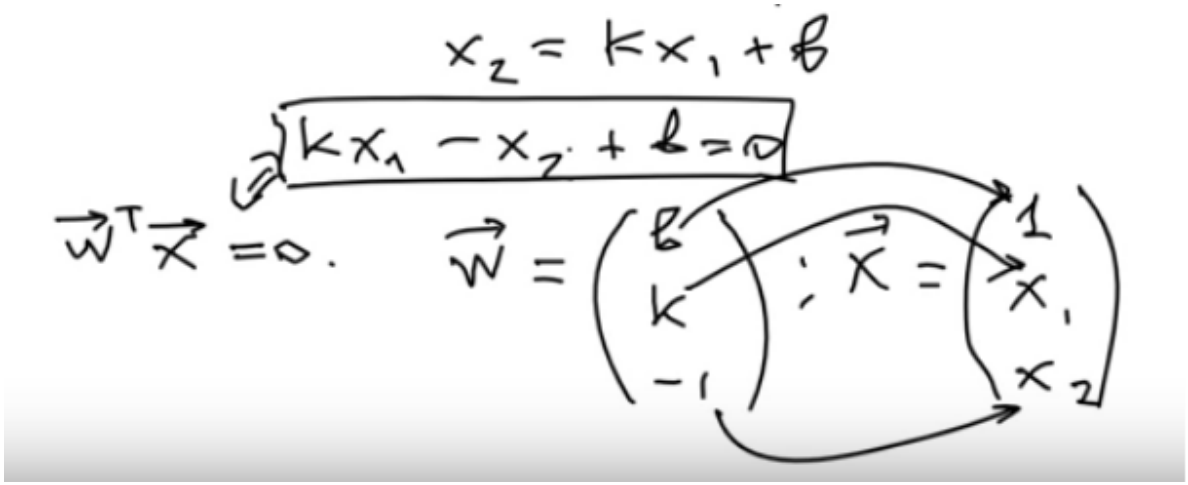
- Let there be 1 input variable x_1 and output variable y, is now called as x_2 (just renaming, for convenience, to be appreciated later !!)
- So equation of line becomes $x_2 = kx_1 + B$ thus $kx_1 - x_2 + B = 0$



(Ref: mlcourse.ai. Lecture 4. Logistic regression. Theory)

Logistic Regression

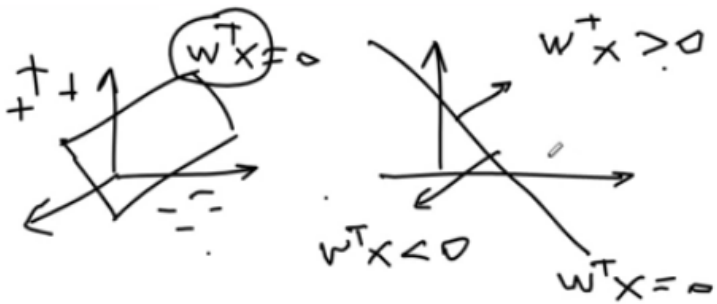
- $kx_1 - x_2 + B = 0$ can be represented as $W^T x = 0$ where $W = [B, k, -1]$ as column vector and $x = [1 x_1 x_2]$ also as column vector.
- The dot product $(3 \times 1)^T \times (3 \times 1) = 1 \times 1$ gives the expression $kx_1 - x_2 + B = 0$
- x can be extended to n dimensions. Then $W^T x = 0$ becomes equation of Hyperplane.



(Ref: mlcourse.ai. Lecture 4. Logistic regression. Theory)

Logistic Regression

- Hyperplane $W^T x = 0$ separates + and - data points, meaning space into 2 regions.
- In one region, for those data points (meaning, x values) $W^T x > 0$ and in the other the data points' x values will evaluate $W^T x < 0$
- Once training is done, W is ready, for a new point, we will evaluate $W^T x$ and see if its less or more or equal to 0 and classify accordingly as $=$ or -.

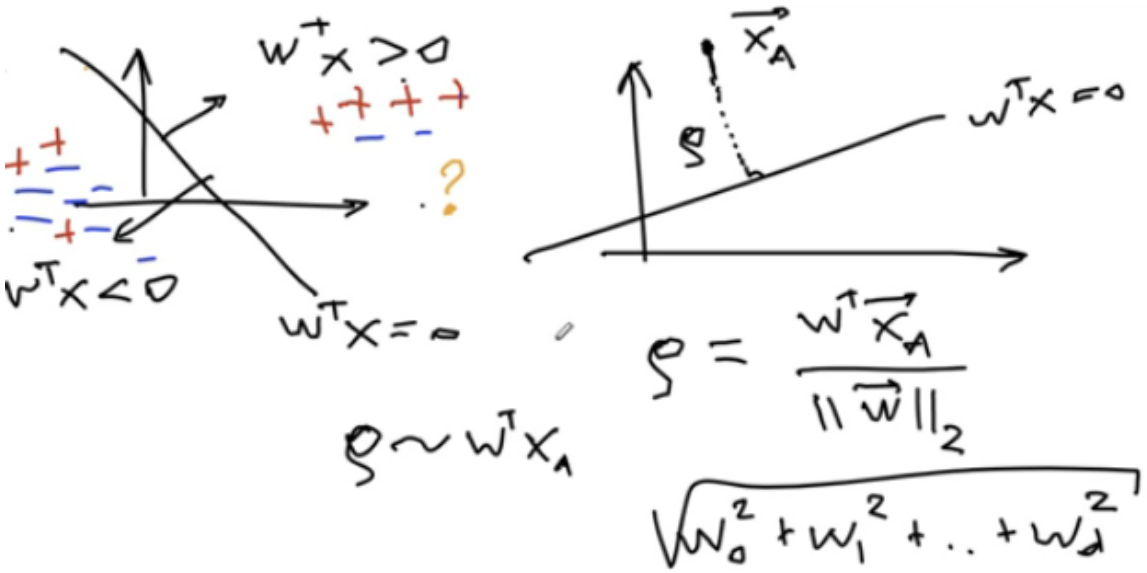


(Note: In some cases data points may be such that, the hyperplane may not be able to separate + and - points perfectly. There will be some mis-classifications. Thats OK.)

(Ref: mlcourse.ai. Lecture 4. Logistic regression. Theory)

Logistic Regression

- $W^T x$ is actually a distance of point A ie x_A from the hyperplane. Just need to divide by norm, so the distance $\rho = \frac{W^T x_A}{||W||}$, where $||W||$ is L_2 norm given by $\sqrt{W_0^2 + W_1^2 + \ldots}$
- ρ can be positive and negative, different for different point. Thus classification.



(Ref: mlcourse.ai. Lecture 4. Logistic regression. Theory)

Logistic Regression - Geometrical Background

- But we don't want to just predict sign (+ or -) but how much + or how much -, ie we want probabilities of the sign.
- Very useful in cases such as Loan customer is good or bad and also how much. We can sort by probabilities and then disburse loan. Probabilities give ratings.

Logistic Regression

So the classifier is $a(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x})$, where

- x – is a feature vector (along with identity);
- w – is a vector of weights in the linear model (with bias w0);
- sign($\bullet$) – is the signum function that returns the sign of its argument;
- $a(x)$ – is a classifier response for x.

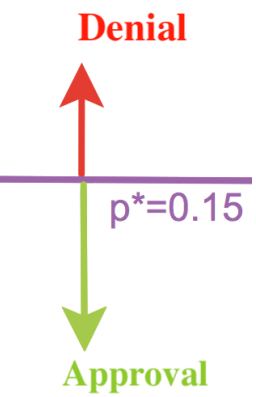
Logistic Regression

- So, Logistic regression is a special case of the linear classifier, but it has an added benefit of predicting a probability p_+ of referring example x_i to the class "+": $p_+ = P(y_i = 1 \mid \mathbf{x}_i, \mathbf{w})$
- Being able to predict not just a response ("+1" or "-1") but the probability of assignment to class "+1" is a very important requirement in many business problems
- E.g. credit scoring where logistic regression is traditionally used.
- Customers who have applied for a loan are ranked based on this predicted probability (in descending order) to obtain a scoreboard that rates customers from bad to good.

Logistic Regression

Below is an example of such a toy scoreboard.

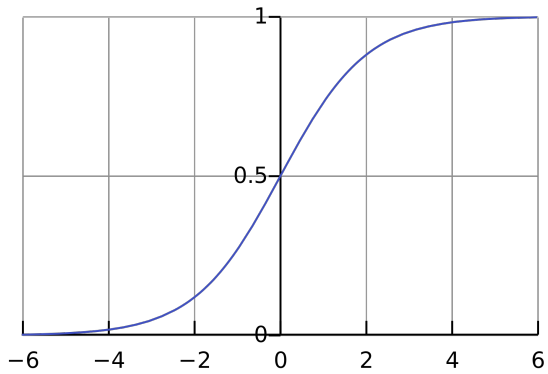
Client	Predicted Default Probability
Mike	0.78
Jack	0.45
Larry	0.13
Kate	0.06
William	0.03
Jessica	0.02



- The bank chooses a threshold p_* to predict the probability of loan default (in the picture it's 0.15) and stops approving loans starting from that value.
- Moreover, it is possible to multiply this predicted probability by the loan amount to get the expectation of losses from the client, which can also constitute good business metrics.

Logistic Regression - Probability calculation

- To predict the probability $p_+ \in [0, 1]$, we can start by constructing a linear prediction using OLS: $b(\mathbf{x}) = \mathbf{w}^\mathsf{T}\mathbf{x} \in \mathbb{R}$. But converting the resulting value to the probability within in the [0, 1] range requires some function $f : \mathbb{R} \to [0, 1]$. Logistic regression uses a specific function for this: $\sigma(z) = \frac{1}{1+\exp^{-z}}$.



Logistic Regression - Probability calculation

Lets take an example:

- Assuming two input variables for simplicity(unlike the 3-dimensional figure shown before)- x_1 and x_2, the function corresponding to the boundary will be something like $\beta_0 + \beta_1 x_1 + \beta_2 x_2$.
- It is crucial to note that x_1 and x_2 are BOTH input variables, and the output variable isn't a part of the conceptual space- unlike a technique like linear regression.

Logistic Regression - Geometrical Background

- Consider a point (a, b). Plugging the values of x_1 and x_2 into the boundary function, we will get its output $\beta_0 + \beta_1 a + \beta_2 b$.
- Now depending on the location of (a, b), there are three possibilities to consider: ...

Logistic Regression - Geometrical Background

I:

- (a, b) lies in the region defined by points of the $+$ class.
- As a result, $\beta_0 + \beta_1 a + \beta_2 b$ will be positive, lying somewhere in $(0, \infty)$.
- Mathematically, the higher the magnitude of this value, the greater is the distance between the point and the boundary.
- Intuitively speaking, the greater is the probability that (a, b) belongs to the $+$ class. Therefore, P_+ will lie in $(0.5, 1]$.

Logistic Regression - Geometrical Background

II:

- (a, b) lies in the region defined by points of the - class.
- Now, $\beta_0 + \beta_1 a + \beta_2 b$ will be negative, lying in $(-\infty, 0)$.
- But like in the positive case, higher the absolute value of the function output, greater the probability that (a, b) belongs to the - class. P_+ will now lie in [0, 0.5).

Logistic Regression - Geometrical Background

III:

- (a, b) lies ON the linear boundary.
- In this case, $\beta_0 + \beta_1 a + \beta_2 b = 0$.
- This means that the model cannot really say whether (a, b) belongs to the $+$ or - class.
- As a result, P_+ will be exactly 0.5.

Logistic Regression - Geometrical Background

- In $a(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x})$, x is a vector like $x = (1, x_1, x_2, \ldots)$, where x_0 is typically taken as a constant, a bias, say, 1. These x features can be, say, Salary, Age, etc.
- W is also set of values like $W = (w_0, w_a, w_2, \ldots)$ these are like coefficients in standard linear equation.
- Resultant is a real number between $-\infty$ to ∞.
- So now we have a function that outputs a value in $(-\infty, \infty)$ given an input data point.
- But how do we map this to the probability P_+, that goes from [0, 1]?
- The answer, is in the odds function (seen before).

Logistic Regression - Geometrical Background

The Trick:

- Let $P(X)$ denote the probability of an event X occurring.
- In that case, the odds ratio ($OR(X)$) is defined by $\frac{P(X)}{1-P(X)}$, which is essentially the ratio of the probability of the event happening, vs. it not happening.
- It is clear that probability and odds convey the exact same information. But as $P(X)$ goes from 0 to 1, $OR(X)$ goes from 0 to ∞.
- However, we are still not quite there yet, since our boundary function gives a value from $-\infty to \infty$. So what we do, is take the logarithm of $OR(X)$, called the log-odds function.
- Mathematically, as $OR(X)$ goes from 0 to ∞, $log(OR(X))$ goes from $-\infty to \infty$!

Logistic Regression - Geometrical Background

The Trick:

- Main idea of the Logistic Regression is to be able to predict $OR(X)$ with just $\mathbf{w}^T\mathbf{x}$
- Now we can say $log(OR_+) = \mathbf{w}^T\mathbf{x}$ as both are from $-\infty to \infty$ and once we equate, we are essentially putting $y = f(x)$, thus, then, the weights could be found out during Training.
- Expanding OR_+ we get $log(\frac{P_+}{1-P_+}) = \mathbf{w}^T\mathbf{x}$
- Taking 'log' to other side as e, the equation becomes
  $P_+ = \frac{e^{\mathbf{w}^T\mathbf{x}}}{1+e^{\mathbf{w}^T\mathbf{x}}} = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$
- This is Sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$

Logistic Regression - Geometrical Background

Finally:

- We have a way to interpret the result of plugging in the attributes of an input into the boundary function.
- The boundary function actually defines the log-odds of the $+$ class, in our model.
- So essentially, in our two-dimensional example, given a point (a, b), this is what Logistic regression would do.

Logistic Regression Steps

Let's see how logistic regression will make a prediction $p_+ = P(y_i = 1 \mid \mathbf{x}_i, \mathbf{w})$. (For now, let's assume that we have somehow obtained weights w i.e. trained the model)

- Step 1. Calculate $w_0 + w_1 x_1 + w_2 x_2 + ... = \mathbf{w}^T \mathbf{x}$. (Equation $\mathbf{w}^T \mathbf{x} = 0$ defines a hyperplane separating the examples into two classes);
- Step 2. Compute the log odds ratio: $\log(OR_+) = \mathbf{w}^T \mathbf{x}$

Logistic Regression Steps

- Step 3. Now that we have the chance of assigning an example to the class of "+" - OR_+, calculate p_+ using the simple relationship:
  $p_+ = \frac{OR_+}{1+OR_+} = \frac{\exp^{\mathbf{w}^T\mathbf{x}}}{1+\exp^{\mathbf{w}^T\mathbf{x}}} = \frac{1}{1+\exp^{-\mathbf{w}^T\mathbf{x}}} = \sigma(\mathbf{w}^T\mathbf{x})$
- On the right side, you can see that we have the sigmoid function.

Logistic Regression Steps

- So, logistic regression predicts the probability of assigning an example to the "+" class (assuming that we know the features and weights of the model) as a sigmoid transformation of a linear combination of the weight vector and the feature vector: $p_+(\mathbf{x}_i) = P(y_i = 1 \mid \mathbf{x}_i, \mathbf{w}) = \sigma(\mathbf{w}^T \mathbf{x}_i)$.
- Next, we will see how the model is trained. We will again rely on maximum likelihood estimation.

Maximum Likelihood Estimation (Advance)

Maximum Likelihood Estimation (MLE) Intuition

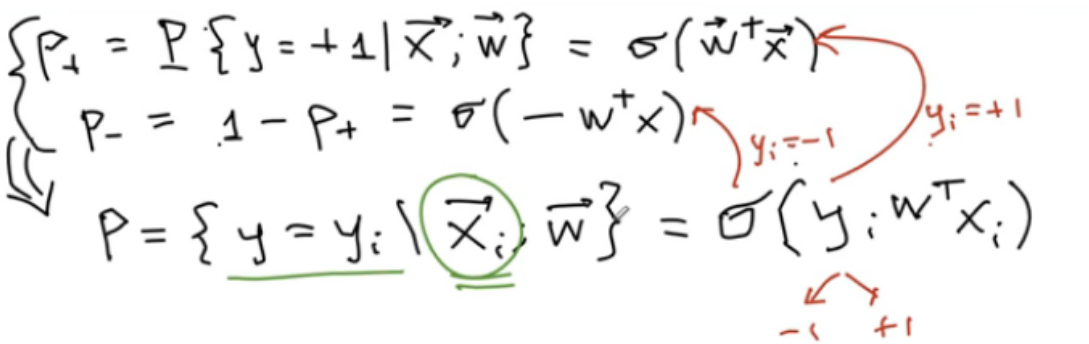
Example

- If you are passing by a Shop everyday at around 2 pm, and you see that the shop is closed (you are in Pune!!!),
- Then maximum likelihood estimate would be that the shop will be closed when you pass by tomorrow 1 pm.
- This MLE approach is often used to fit distributions in observed data.
- Distribution is what you see when you plot Histogram of observed data, ie data values on x axis and frequencies of occurrences of each data values on y axis.
- Once you fit the distribution, say Normal Distribution, you get parameters μ and σ

Maximum Likelihood Estimation (MLE) Intuition

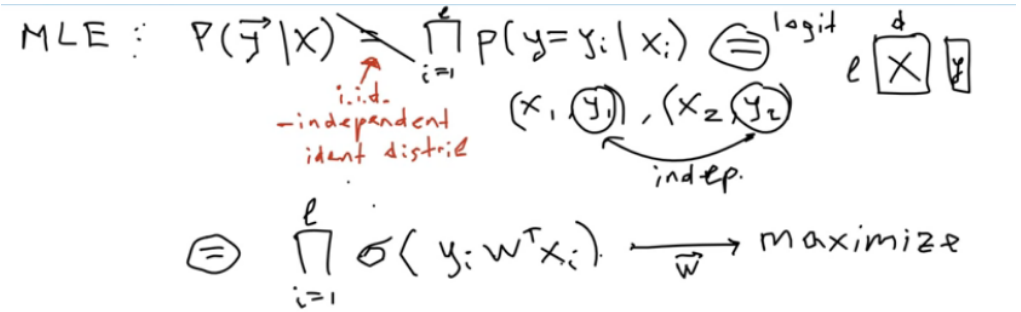
In logistic regression, we will estimate W's with MLE. How?

- $P = \{y = y_i | x_i; W\} = Sigmoid(y_i W^T x_i)$ is a cleaver combination for $y = \pm 1$
- MLE is for maximizing probability of observing vector y given that we observer matrix X and parameters matrix W.



Maximum Likelihood Estimation (MLE) Intuition

- MLE Probability can be expressed as $P(y|x) = \Pi_{i=1}^{l} P(y = y_i|x_i)$, where l are number of rows and d columns/features.
- Assumption: All observations (say, $(x1, y_1), (x_2, y_z) \dots$) are IID (Independently Identically Distributed) data.
- The probability can be equated in terms of Sigmoid, as seen before. $= \Pi_{i=1}^{l} \sigma(y_i W^T X - i)$. We will maximize this.
- Instead of maximizing product, its better to maximize sum. so we take logs, then it becomes a sum, them maximize it. (Note: 'log' being monotonic,does not affect maximization)

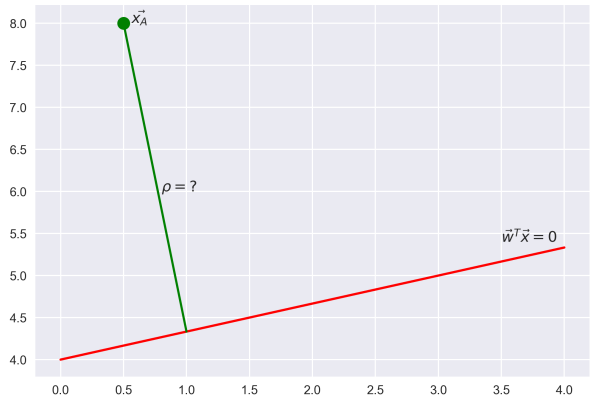


Maximum Likelihood Estimation and Logistic Regression

- Let's see how an optimization problem for logistic regression is obtained from the MLE, namely, minimization of the logistic loss function.
- We have just seen that logistic regression models the probability of assigning an example to the class "+" as:
  $p_+(\mathbf{x}_i) = P(y_i = 1 \mid \mathbf{x}_i, \mathbf{w}) = \sigma(\mathbf{w}^T \mathbf{x}_i)$
- Then, for the class "-", the corresponding expression is as follows:
  $p_-(\mathbf{x}_i) = P(y_i = -1 \mid \mathbf{x}_i, \mathbf{w}) = 1 - \sigma(\mathbf{w}^T \mathbf{x}_i) = \sigma(-\mathbf{w}^T \mathbf{x}_i)$
- Both of these expressions can be cleverly combined into one (watch carefully, maybe you are being tricked): $P(y = y_i \mid \mathbf{x}_i, \mathbf{w}) = \sigma(y_i \mathbf{w}^T \mathbf{x}_i)$

Geometrical Interpretation

- It is known (or given or assumed!!) find the distance from the point with a radius-vector x_A to a plane defined by the equation $\mathbf{w}^T\mathbf{x} = 0$
- Answer: $\rho(\mathbf{x}_A, \mathbf{w}^T\mathbf{x} = 0) = \frac{\mathbf{w}^T\mathbf{x}_A}{||\mathbf{w}||}$



- The greater the absolute value of the expression $\mathbf{w}^T\mathbf{x}_i$, the farther the point x_i is from the plane $\mathbf{w}^T\mathbf{x} = 0$

Maximum Likelihood Estimation and Logistic Regression

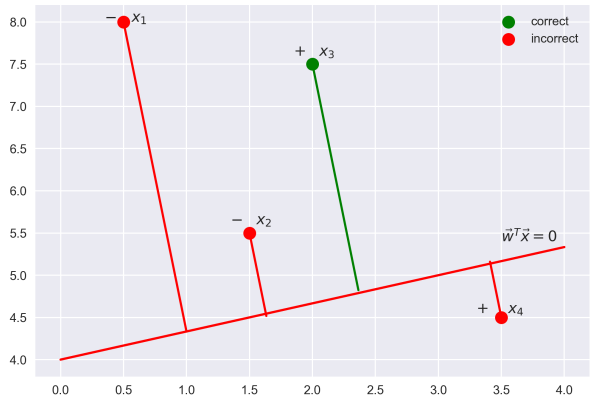
Trick:

- In the expression $M(\mathbf{x}_i) = y_i \mathbf{w}^T \mathbf{x}_i$, the $\mathbf{w}^T \mathbf{x}_i$ part is for the distance of x from the hyperplane, but if we multiply it by y then it is known as the margin of classification on the object x_i (not to be confused with a gap, which is also called margin, in the SVM context).
- If it is non-negative, the model is correct in choosing the class of the object x_i; if it is negative, then the object x_i is misclassified.
- Note that the margin is defined for objects in the training set only where real target class labels y_i are known.
- In following figures, green is correct whereas red is not.

Maximum Likelihood Estimation and Logistic Regression

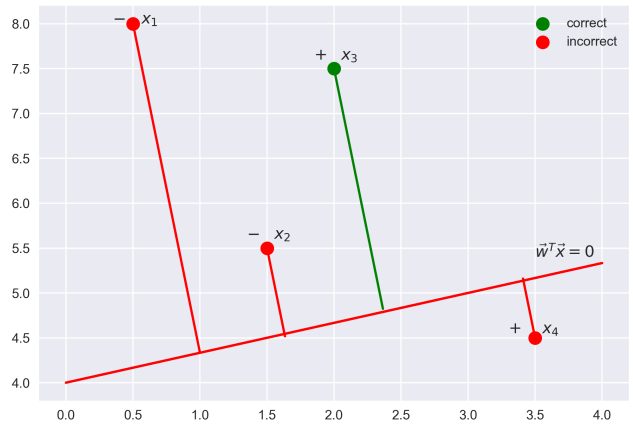
Hence, our expression $M(\mathbf{x}_i) = y_i \mathbf{w}^T \mathbf{x}_i$ is a kind of "confidence" in our model's classification of the object x_i:

- If the margin is large (in absolute value) and positive, the class label is set correctly, and the object is far away from the separating hyperplane i.e. classified confidently. See Point x3 on the picture;

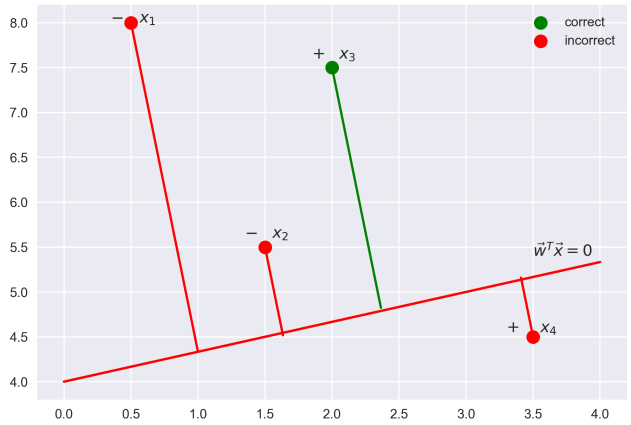


Maximum Likelihood Estimation and Logistic Regression

If the margin is large (in absolute value) and negative, then class label is set incorrectly, and the object is far from the separating hyperplane (the object is most likely an anomaly; for example, it could be improperly labeled in the training set). See Point x_1 on the picture;

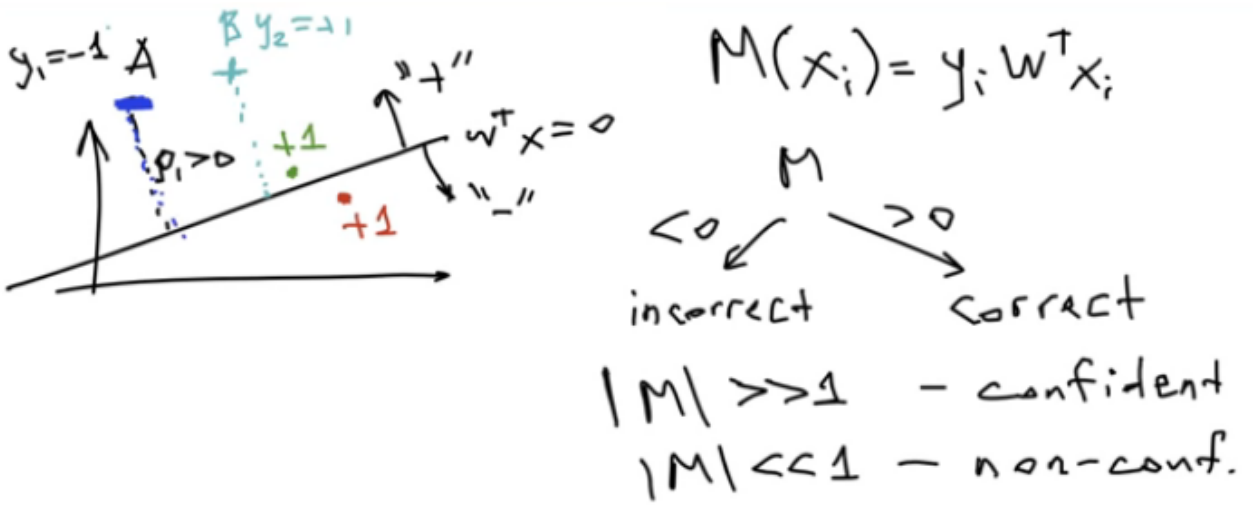


Maximum Likelihood Estimation and Logistic Regression



If the margin is large (in absolute value) and negative, then class label is set incorrectly, and the object is far from the separating hyperplane (the object is most likely an anomaly; for example, it could be improperly labeled in the training set). See Point x_1 on the picture;

Maximum Likelihood Estimation and Logistic Regression



We are now maximizing sigmoid of Margins wrt W, meaning a big sum of logs with exponential functions inside. We convert maximization to minimization with negative sign removed.

$$\sum_{i=1}^{\ell} \log \sigma(y_i w^T x_i) = \sum_{i=1}^{\ell} \log\left(1+e^{\overbrace{-y_i w^T x_i}^{M_i}}\right) \to \min$$

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

MLE $\vec{w} \to \vec{w}^*$

Maximum Likelihood Estimation and Logistic Regression

We use some Gradient Descent or something to get good Ws.

$$\#\text{mistakes} = \frac{1}{\ell}\sum_{i=1}^{\ell}[y_i \neq a(x_i)] = \frac{1}{\ell}\sum_{i=1}^{\ell}[M(x_i) < 0]$$

Maximum Likelihood Estimation and Logistic Regression

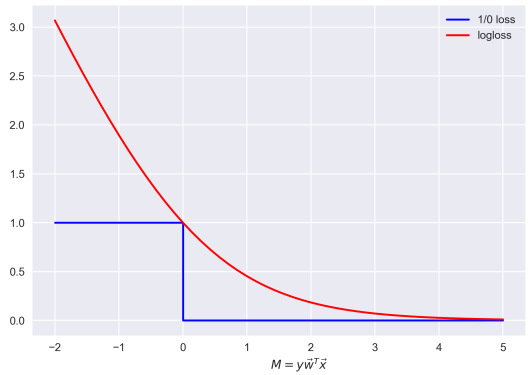
- Let's now compute the likelihood of the data set i.e. the probability of observing the given vector y from data set X. We'll make a strong assumption: objects come independently from one distribution (i.i.d.). Then, we can write $P(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \prod_{i=1}^{\ell} P(y = y_i \mid \mathbf{x}_i, \mathbf{w})$, where ℓ is the length of data set X (number of rows).
- As usual, let's take the logarithm of this expression because a sum is much easier to optimize than the product:
$\log P(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \log \prod_{i=1}^{\ell} P(y = y_i \mid \mathbf{x}_i, \mathbf{w}) = \log \prod_{i=1}^{\ell} \sigma(y_i \mathbf{w}^{\mathsf{T}} \mathbf{x}_i) =$

Maximum Likelihood Estimation and Logistic Regression

- $= \sum_{i=1}^{\ell} \log \sigma(y_i \mathbf{w}^T \mathbf{x}_i) = \sum_{i=1}^{\ell} \log \frac{1}{1+\exp^{-y_i \mathbf{w}^T \mathbf{x}_i}} = -\sum_{i=1}^{\ell} \log(1+\exp^{-y_i \mathbf{w}^T \mathbf{x}_i})$
- Maximizing the likelihood is equivalent to minimizing the expression: $\mathcal{L}_{\log}(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \sum_{i=1}^{\ell} \log(1 + \exp^{-y_i \mathbf{w}^T \mathbf{x}_i})$
- This is logistic loss function that is summed over all objects in the training set.

Maximum Likelihood Estimation and Logistic Regression

- Let's look at the new function as a function of margin $L(M) = \log(1 + \exp^{-M})$ and plot it along with zero-one loss graph, which simply penalizes the model for error on each object by 1 (negative margin): $L_{1/0}(M) = [M < 0]$



- The picture reflects the idea that, if we are not able to directly minimize the number of errors in the classification problem (at least not by gradient methods - derivative of the zero-one loss function at zero turns to infinity), we can minimize its upper bounds.

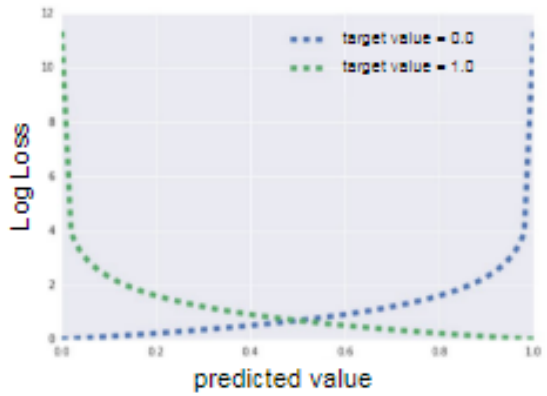
Maximum Likelihood Estimation and Logistic Regression

- For the logistic loss function (where the logarithm is binary, but this does not matter), the following is valid: $\mathcal{L}_{\infty/\prime}(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \sum_{i=1}^{\ell} [M(\mathbf{x}_i) < 0] \leq \sum_{i=1}^{\ell} \log(1 + \exp^{-y_i \mathbf{w}^T \mathbf{x}_i}) = \mathcal{L}_{\log}(\mathbf{X}, \mathbf{y}, \mathbf{w})$,
- where $\mathcal{L}_{\infty/\prime}(\mathbf{X}, \mathbf{y})$ is simply the number of errors of logistic regression with weights w on a data set $(\mathbf{X}, \mathbf{y})$.
- Thus, by reducing the upper bound of $\mathcal{L}_{\updownarrow\wr\}}$ by the number of classification errors, we hope to reduce the number of errors itself.

LogLoss Defined

Instead of Squared-Loss defined for Linear Regression, we have Log-Loss defined for Logistic Regression.

$$LogLoss = \sum_{(x,y)\in D} -y\,log(y') - (1-y)\,log(1-y')$$



Loss is most near 0 or 1. We are trying to minimize it.

LogLoss Rational : Maximum Likelihood Estimation

- For a survey on whether India will win the next world cup, 400 answered.
- 117 people were Positive.
- Then, it is reasonable to assume that the probability that the next respondent feels that India will win, is $\frac{117}{400} \approx 29\%$
- This intuitive assessment is not only good but also a maximum likelihood estimate.

Background: Bernoulli distribution

- Bernoulli distribution: a random variable has a Bernoulli distribution if it takes only two values (1 and 0 with probability θ and $1-\theta$ respectively) and has the following probability distribution function:
  $p(\theta, x) = \theta^x (1-\theta)^{(1-x)}, x \in \{0, 1\}$
- This distribution is exactly what we need, and the distribution parameter θ is the estimate of the probability that a person feels India will win.

Background: Bernoulli distribution

- In our 400 independent experiments, let's denote their outcomes as $\mathbf{x} = (x_1, x_2, \ldots, x_{400})$
- Let's write down the likelihood of our data (observations), i.e. the probability of observing exactly these 117 realizations of the random variable $x = 1$ and 283 realizations of $x = 0$:
$p(\mathbf{x}; \theta) = \prod_{i=1}^{400} \theta^{x_i} (1-\theta)^{(1-x_i)} = \theta^{117} (1-\theta)^{283}$

Background: Bernoulli distribution

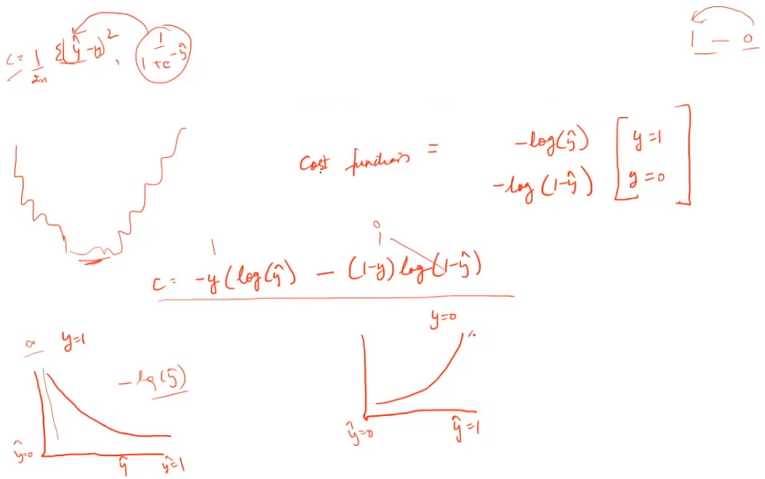
- Next, we will maximize the expression with respect to θ. Most often this is not done with the likelihood $p(\mathbf{x};\theta)$ but with its logarithm (the monotonic transformation does not change the solution but simplifies calculation greatly): $\log p(\mathbf{x};\theta) = \log \prod_{i=1}^{400} \theta^{x_i} (1-\theta)^{(1-x_i)} =$
$= \log \theta^{117} (1-\theta)^{283} = 117 \log \theta + 283 \log (1-\theta)$
- Now, we want to find such a value of θ that will maximize the likelihood.

Background: Bernoulli distribution

- For this purpose, we'll take the derivative with respect to θ, set it to zero, and solve the resulting equation:
$\frac{\partial \log p(\mathbf{x};\theta)}{\partial \theta} = \frac{\partial}{\partial \theta}\left(117 \log \theta + 283 \log (1-\theta)\right) = \frac{117}{\theta} - \frac{283}{1-\theta}$;
- It turns out that our intuitive assessment is exactly the maximum likelihood estimate.

Cost function of logistic regression

- Cost function for Linear Regression is $\frac{1}{2m}\sum(\hat{y}-y)^2$
- Predicted probability in Logistic Regression depends on Sigmoid $P = \frac{1}{1+e^{-\hat{y}}}$
- Putting this P into Cost function of Linear Regression will turn the cost function into some complex one, for which finding minima would be difficult.
- We would be using 2 cost functions, one for $y = 1$ and another for $y = 0$. Both can be combined into one.
- plot of cost function, for actual $y = 1$ shows that when predicted is 1 and actual is one, the cost is minimal (tail on right). When predicted is 0 for this y=1 curve, the cost is infinity (left top)



Cost function of logistic regression

- We have a dataset X consisting of m data-points and n features. And there is a class variable y a vector of length m which can have two values 1 or 0.
- Now logistic regression says that the probability that class variable value yi=1 , i=1,2,...m can be modeled as follows:
  $$P(y_i = 1|\mathbf{x}_i; \theta) = h_\theta(\mathbf{x}_i) = \frac{1}{1 + e^{(-\theta^T \mathbf{x}_i)}}$$
- So $y_i = 1$ with probability $h_\theta(\mathbf{x}_i)$ and $y_i = 0$ with probability $1 - h_\theta(\mathbf{x}_i)$.

(Ref: Deriving cost function using MLE :Why use log function? - Stack Overflow)

Cost function of logistic regression

- This can be combined into a single equation as follows, (actually y_i follows a Bernoulli distribution) $P(y_i) = h_\theta(\mathbf{x}_i)^{y_i}(1 - h_\theta(\mathbf{x}_i))^{1-y_i}$
- $P(y_i)$ is known as the likelihood of single data point x_i, i.e. given the value of y_i what is the probability of x_i occurring. it is the conditional probability $P(\mathbf{x}_i|y_i)$.
- The likelihood of the entire dataset X is the product of the individual data point likelihoods. Thus
  $P(\mathbf{X}|\mathbf{y}) = \prod_{i=1}^{m} P(\mathbf{x}_i|y_i) = \prod_{i=1}^{m} h_\theta(\mathbf{x}_i)^{y_i}(1 - h_\theta(\mathbf{x}_i))^{1-y_i}$

(Ref: Deriving cost function using MLE :Why use log function? - Stack Overflow)

Cost function of logistic regression

- Now the principle of maximum likelihood says that we find the parameters that maximise likelihood $P(\mathbf{X}|\mathbf{y})$.
- logarithms are used because they convert products into sums and do not alter the maximization search, as they are monotone increasing functions. Here too we have a product form in the likelihood.So we take the natural logarithm as maximising the likelihood is same as maximising the log likelihood, so log likelihood $L(\theta)$ is now:
  $L(\theta) = \log(P(\mathbf{X}|\mathbf{y}) = \sum_{i=1}^{m} y_i \log(h_\theta(\mathbf{x}_i)) + (1 - y_i) \log(1 - h_\theta(\mathbf{x}_i))$

(Ref: Deriving cost function using MLE :Why use log function? - Stack Overflow)

Cost function of logistic regression

- Since in linear regression we found the θ that minimizes our cost function , here too for the sake of consistency, we would like to have a minimization problem. And we want the average cost over all the data points.
- Currently, we have a maximization of $L(\theta)$. Maximization of $L(\theta)$ is equivalent to minimization of $-L(\theta)$. And using the average cost over all data points, our cost function for logistic regression comes out to be,

$$J(\theta) = -\frac{1}{m} L(\theta)$$
$$= -\frac{1}{m} \left(\sum_{i=1}^{m} y_i \log(h_\theta(\mathbf{x}_i)) + (1 - y_i) \log(1 - h_\theta(\mathbf{x}_i)) \right)$$

(Ref: Deriving cost function using MLE :Why use log function? - Stack Overflow)

Cost function of logistic regression

- Now we can also understand why the cost for single data point comes as follows:
- the cost for a single data point is $= -\log(P(\mathbf{x}_i|y_i))$ which can be written as $-(y_i \log(h_\theta(\mathbf{x}_i)) + (1-y_i)\log(1-h_\theta(\mathbf{x}_i))$
- We can now split the above into two depending upon the value of y_i. Thus we get $J(h_\theta(\mathbf{x}_i), y_i) = -\log(h_\theta(\mathbf{x}_i)),$ if $y_i = 1$ and $J(h_\theta(\mathbf{x}_i), y_i) = -\log(1-(h_\theta(\mathbf{x}_i)),$ if $y_i = 0$

(Ref: Deriving cost function using MLE :Why use log function? - Stack Overflow)

Evaluation Metrics

Accuracy

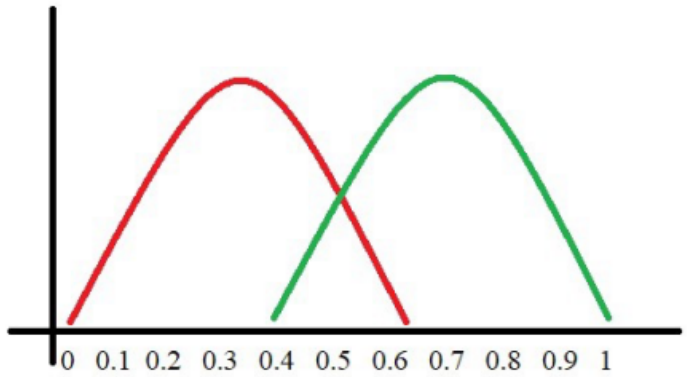
- How do we evaluate classification models?
- One possible measure: Accuracy
  - the fraction of predictions we got right
- Accuracy Can Be Misleading

Accuracy

- In case of cancer dataset, if you blatantly predict "No Cancer" you are going to be right 99% times, as thats the percent of Non-Cancer patients in the population.
- In this case, such great accuracy is actually useless, as the class/target is imbalanced.
- Need Confusion Matrix.

What is ROC?

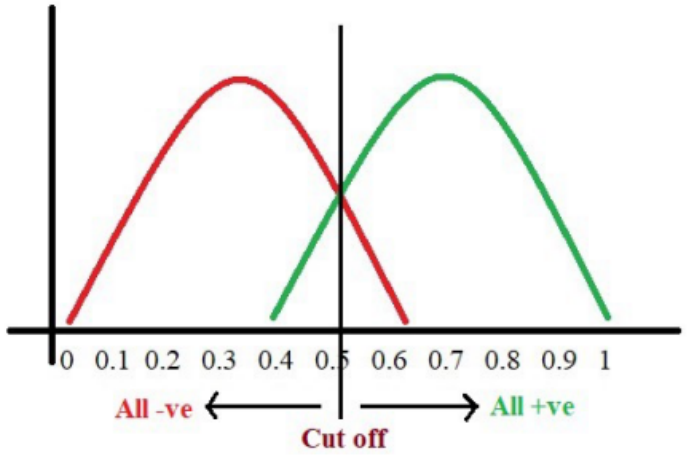
- ROC : Receiver Operating Characteristics Curve
- ROC tells us about how good the model can distinguish between two things (e.g If a patient has a disease or no).
- Better models can accurately distinguish between the two.
- Predictions are plotted on x, and colors show True or false. Y is frequency or counts.

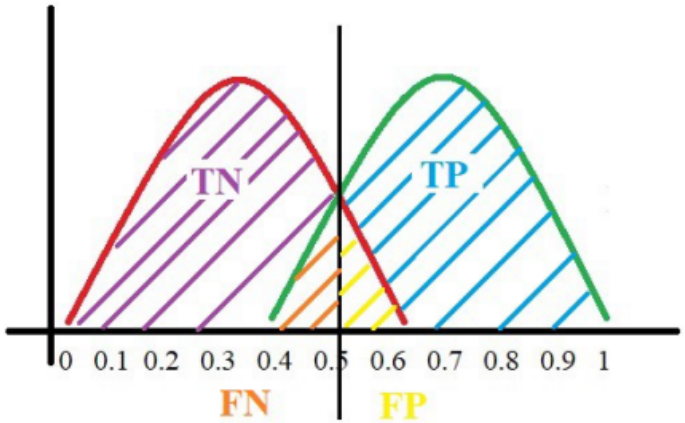


(Ref: Let's learn about AUC ROC Curve! – GreyAtom)

What is ROC?

- Pick a value where we need to set the cut off say, "0.5" as shown
- All the positive values above the threshold will be "True Positives"
- Negative values above the threshold will be "False Positives" as they are predicted incorrectly as positives.
- All the negative values below the threshold will be "True Negatives"
- Positive values below the threshold will be "False Negative" as they are predicted incorrectly as negatives.

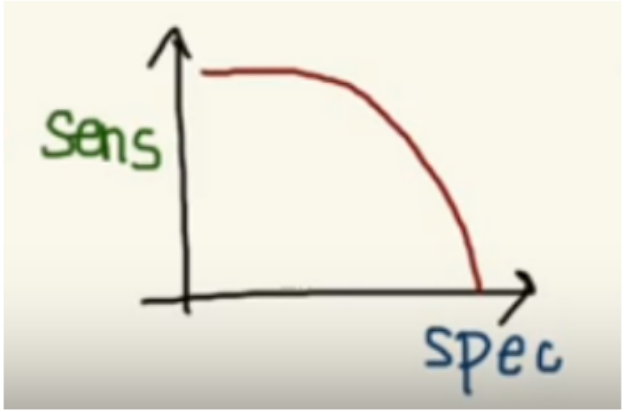




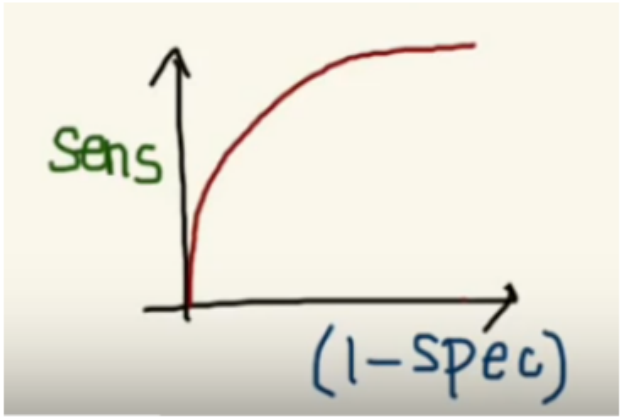
(Ref: Let's learn about AUC ROC Curve! – GreyAtom)

What is Sensitivity and Speci4city?

- Sensitivity or Recall = TP/TP + FN.
- Specificity = TN/TN + FP.
- When we decrease the threshold, we get more positive values thus increasing the sensitivity. Meanwhile, this will decrease the specificity.
- and vice versa. Meaning, there is a trade-off.
- Instead of Specificity we use (1 - Specificity) and the graph will look something like the bottom one:



Trade off between Sensitivity & Specificity



(Ref: Let's learn about AUC ROC Curve! – GreyAtom)

Why do we use (1- Specificity)?

- Let's derive what exactly is (1-Specificity)
- Specificity gives us the True Negative Rate
- (1 - Specificity) gives us the False Positive Rate Rate
- So now we are just looking at the positives.
- As we increase the threshold, we decrease the TPR as well as the FPR and when we decrease the threshold, we are increasing the TPR and FPR.

$$Specificity = \frac{TN}{TN + FP}$$

$$1 - Specificity = 1 - \frac{TN}{TN + FP}$$

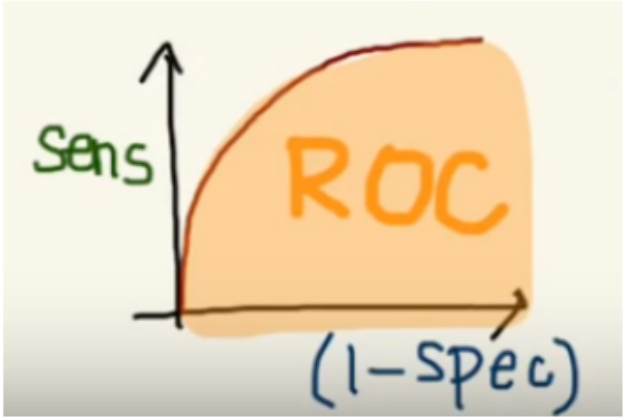
$$1 - Specificity = \frac{TN + FP - TN}{TN + FP}$$

$$1 - Specificity = \frac{FP}{TN + FP}$$

(Ref: Let's learn about AUC ROC Curve! – GreyAtom)

What is AUC?

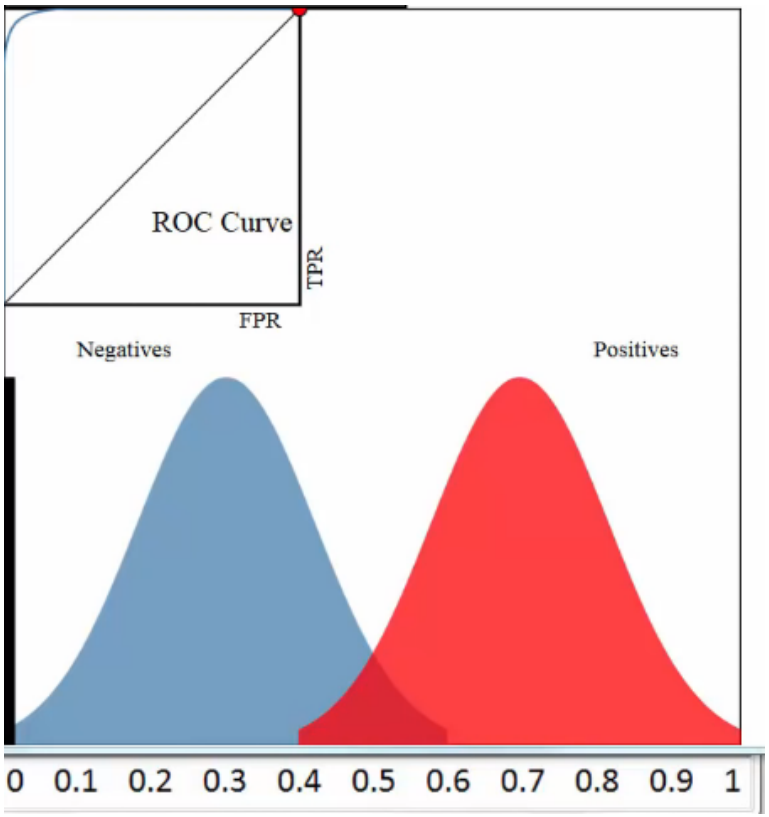
- The AUC is the area under the ROC curve.
- It gives us a good idea of how well the model performances.
- AUC ROC indicates how well the probabilities from the positive classes are separated from the negative classes.



(Ref: Let's learn about AUC ROC Curve! – GreyAtom)

ROC AUC Intuition (More Details)

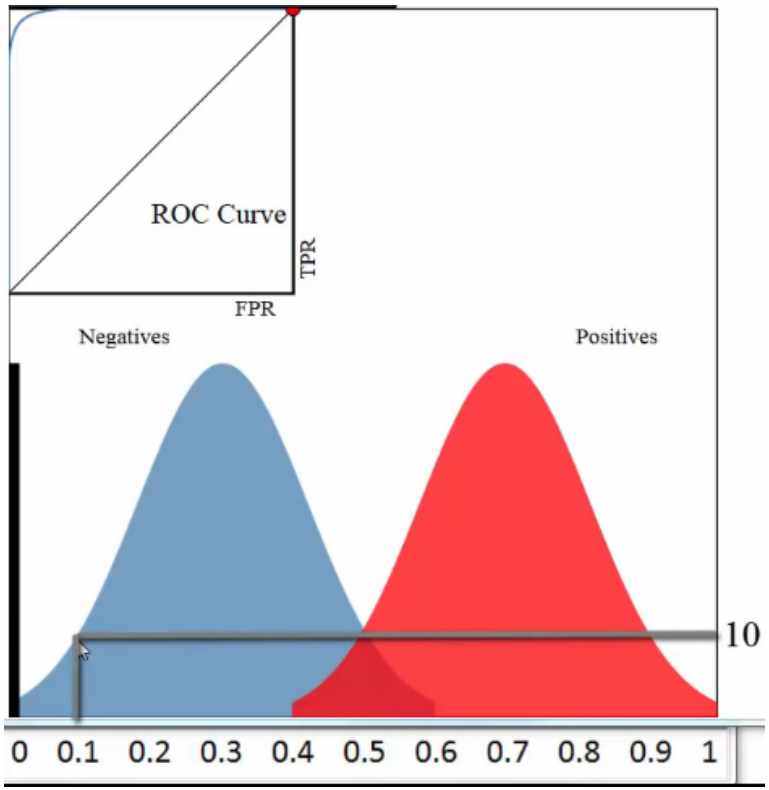
- ROC : Receiver Operating Characteristics Curve
- AUC: Area under curve
- Often used to evaluate performance of a binary classifier
- Two possible outcomes: Positive and Negative
- This is histogram (continuous) where red pixels are positive (got admission to IIT!!) and blue pixels are negative (did not get admitted)
- 250 students (pixels here, for example) got admitted and 250 did not get admitted.



(Ref: ROC Curves and Area Under the Curve (AUC) Explained - Data School)

ROC AUC Intuition

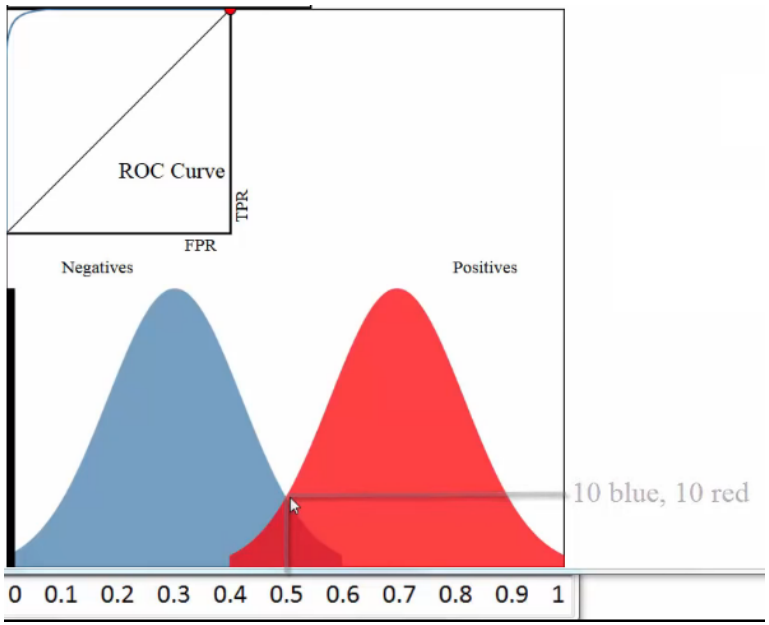
- This is result of predictions done on validation set. For which, you have actual values as well.
- The classifier is like Logistic Regression where you get probabilities as well.
- Predicted probabilities are seen on the x axis. Y axis is count of students with ACTUAL red(admitted) or blue (not admitted) values
- E.g. there were 10 students for which you predicted admission probability of 0.1, they did not get admitted shown as negative (blue)



(Ref: ROC Curves and Area Under the Curve (AUC) Explained - Data School)

ROC AUC Intuition

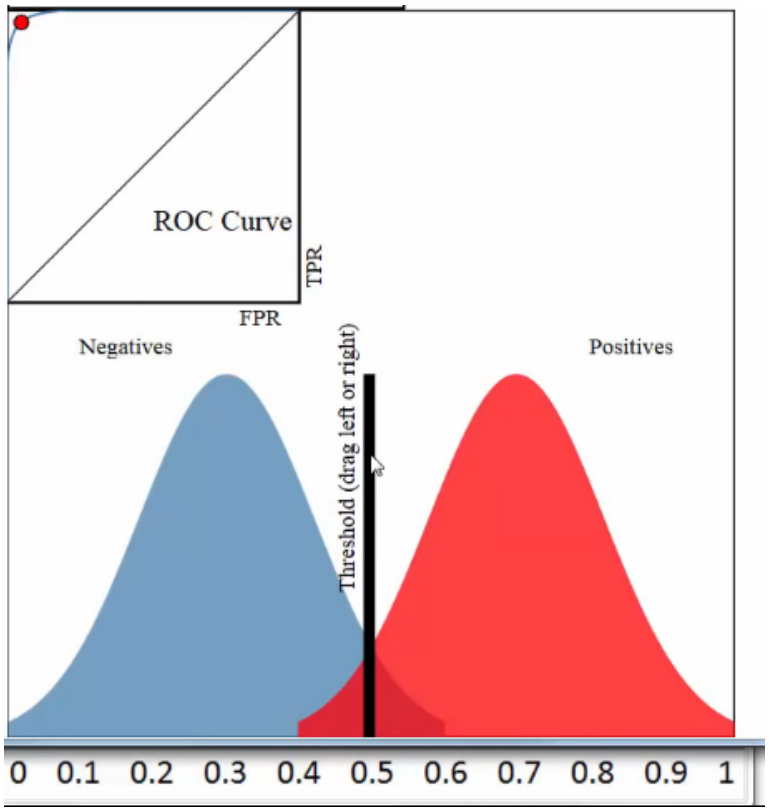
- E.g. there were 50 students for which you predicted admission probability of 0.3, they did not get admitted, ie negative (blue) (not shown here, but height of the bell curve is 50)
- For 20 students where you predicted 0.5 probability of admission, 10 did not get admitted (blue) but 10 got admitted (red)
- You can say that the classifier is doing well as it is able to partition/separate two classes reasonably well (only small overlap or ambiguity in the middle)



(Ref: ROC Curves and Area Under the Curve (AUC) Explained - Data School)

ROC AUC Intuition

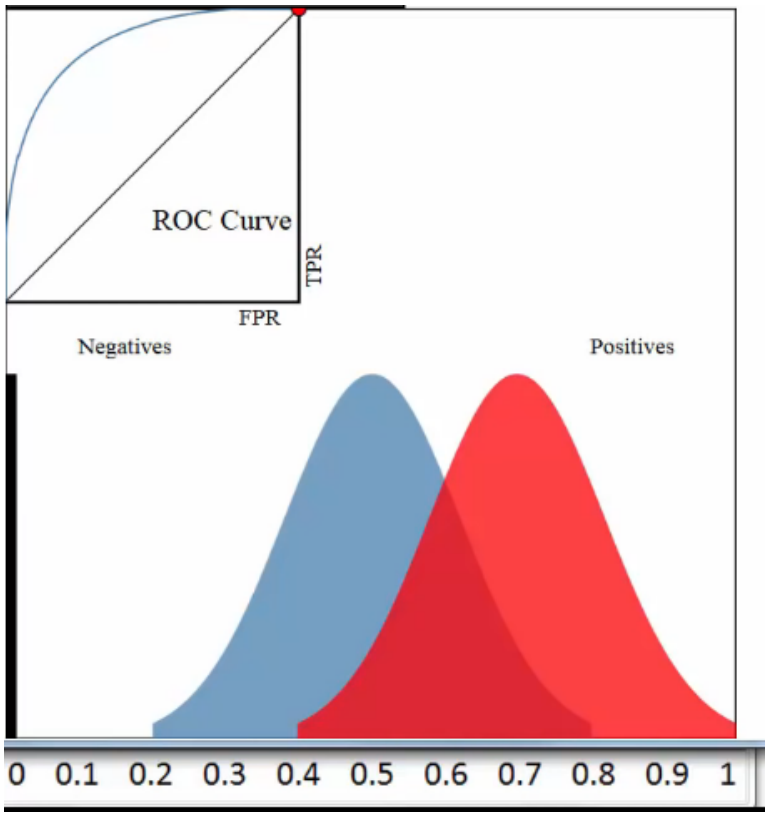
- So. we can put threshold (or the partition) at 0.5, anything about that is admitted, and below is not admitted.
- With this threshold the accuracy looks to be in the range of 90%.
- Please not that the ROC curve in the top left is almost touching the left and top boundaries. Its bit away at the corner due to some errors in the middle region.



(Ref: ROC Curves and Area Under the Curve (AUC) Explained - Data School)

ROC AUC Intuition

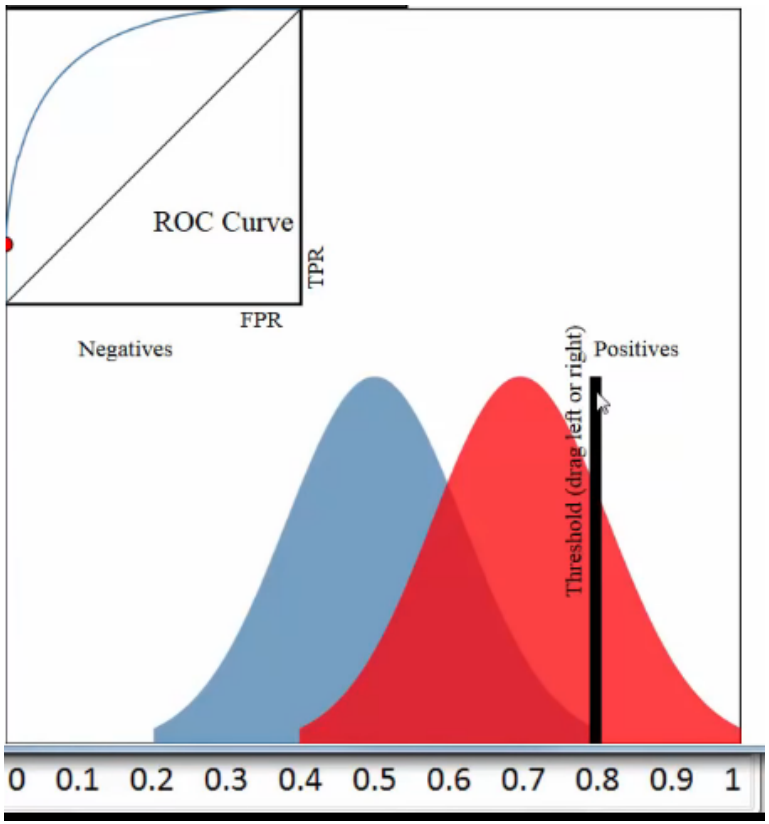
- If the classifier was not good, there would be far more overlap in the middle,ie many misclassifications.
- Regardless of where you put threshold, your classification accuracy will be bad
- Note that the ROC curve has shifted and too much gap way from the top left corner, so many errors.
- ROC curve is a plot of True Positive Rate (TPR) on y axis and False Positive Rate (FPR) on x axis for every possible classification threshold (say, 0 to 1)



(Ref: ROC Curves and Area Under the Curve (AUC) Explained - Data School)

ROC AUC Intuition

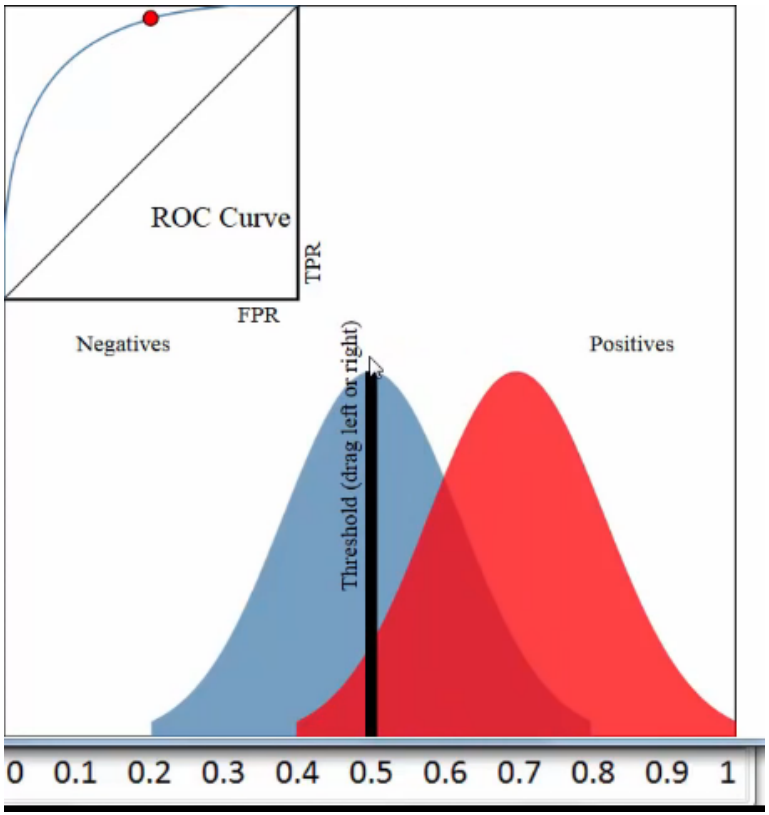
- TPR : True Positives Predicted divided by All actual positives
- FPR : False Positives Predicted divided by All actual negatives
- Both range from 0 to 1.
- For threshold 0.8, TPR is pixels on right (approx 50) divided by all reds (actual positives) ie 250, so $50/250 = 0.2$,
- FPR is blue pixels on right of the line (0) divided by all blues ie 250, so $0/250 = 0$
- Plot point at $x = 0$, and $y = 0.2$



(Ref: ROC Curves and Area Under the Curve (AUC) Explained - Data School)

ROC AUC Intuition

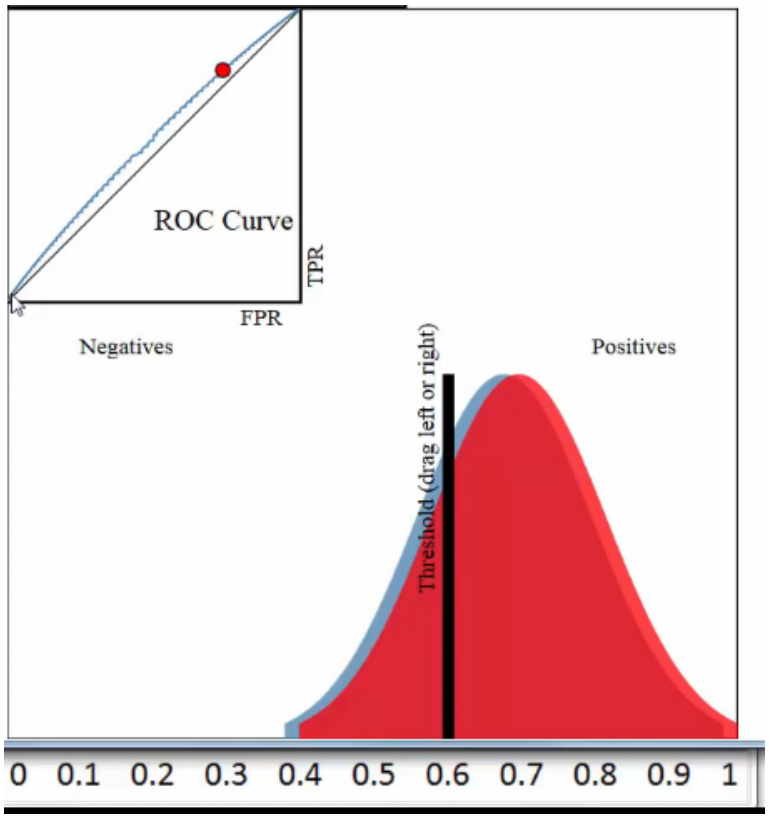
- For threshold 0.5, TPR is pixels on right (approx 230) divided by all reds (actual positives) ie 250, so $230/250 = 0.94$,
- FPR is blue pixels on right of the line (125) divided by all blues ie 250, so $125/250 = 0.5$
- Plot point at x $= 0.5$, and y $=$ 0.94



(Ref: ROC Curves and Area Under the Curve (AUC) Explained - Data School)

ROC AUC Intuition

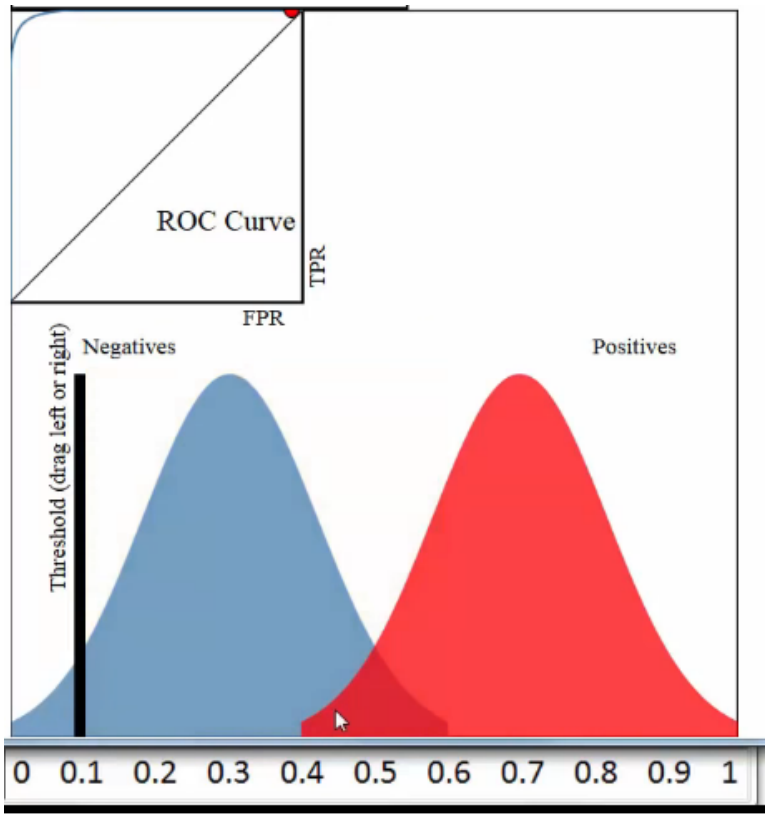
- Similarly plot for all possible thresholds.
- Thus ROC curve visualizes all possible thresholds.
- It shows how the classifier is AWAY from the ideal, top left corner
- The worst classier would be the diagonal straight line. 50% line ie random chance.
- So, to distinguish good classifier from a bad one, Area Under Curve is used.



(Ref: ROC Curves and Area Under the Curve (AUC) Explained - Data School)

ROC AUC Intuition

- AUC is Area under ROC curve divided by the whole square, say 0.8 for this example.
- A poor classifier (diagonal line) will have area 0.5
- Best classifier will be (corner legs) will have area 1.
- Note that the Class labels are balanced in this example, ie number of students admitted and not admitted is roughly equal), so heights of the bell curves are same. That does not affect ROC AUC calculations though.
- Choosing threshold is a business decision.



(Ref: ROC Curves and Area Under the Curve (AUC) Explained - Data School)

ROC AUC

- ROC AUC curve can not be used to compare two models.
- As it deals with only the order of probabilities.
- So, even though model looks far better than model 2, both will have same ROC curve and thus same AUC.
- A better criteria for comparison would be log-loss.

ID	Actual	Predicted probabilities 1	Predicted probabilities 2	Predicted class
ID6	1	0.94	0.74	1
ID1	1	**0.90**	**0.69**	**1**
ID7	1	0.78	0.63	0
ID8	0	0.56	0.57	0
ID2	0	0.51	0.51	0
ID3	1	0.47	0.44	0
ID4	1	0.32	0.39	0
ID5	0	0.10	0.35	0

(Ref: Introduction to Data Science - Analytics Vidhya)

Log Loss

- As predictions are for output “1”, probabilities are adjusted first.
- Then the diff (loss) is computed. Its log is taken.
- As log is negative, just to make it positive -ve sign is added.
- This procedure can be concisely put in a formula, as shown

ID	Actual	Predicted probabilities	Corrected Probabilities	Log
ID6	1	0.94	0.94	-0.0268721464
ID1	1	0.90	0.90	-0.0457574906
ID7	1	0.78	0.78	-0.1079053973
ID8	0	0.56	0.44	-0.3565473235
ID2	0	0.51	0.49	-0.30980392
ID3	1	0.47	0.47	-0.3279021421
ID4	1	0.32	0.32	-0.4948500217
ID5	0	0.10	0.90	-0.0457574906

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

(Ref: Introduction to Data Science - Analytics Vidhya)

Complex Variations

Multiple Xs Logistic Regression

- How to prediction a binary response using multiple predictors?
- Can generalize the logistic function to p predictors

$$p(X) = \frac{e^{\beta_0+\beta_1 X+\ldots+\beta_p X}}{1+e^{\beta_0+\beta_1 X+\ldots+\beta_p X}} \qquad \log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0+\beta_1 X_1+\ldots+\beta_p X_p$$

Multi Class Classification

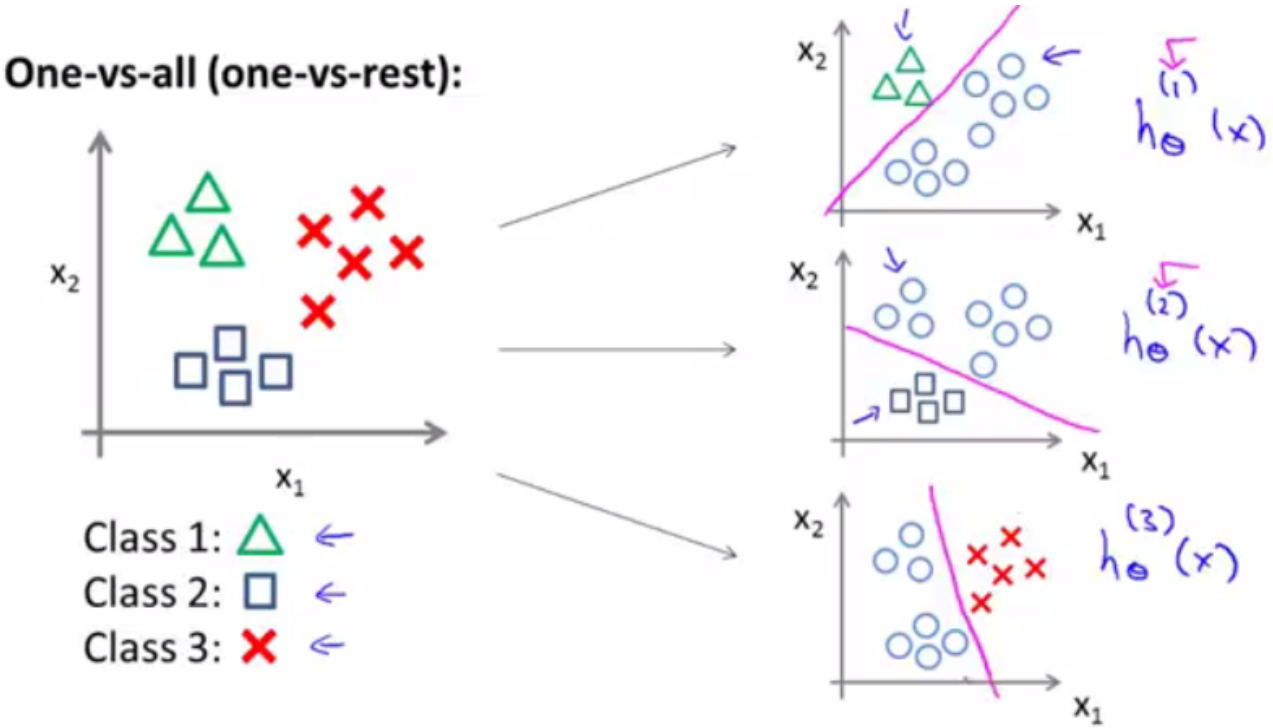
- Example: Email tagging: Work, Friends, Family, Hobby.
- 4 class problem: y = 1 or 2 or 3 or 4
- Other Example: Weather: Sunny, Cloudy, Rainy, Snow
- Target has discrete values (int)

Multi Class Logistic Regression

- How to prediction a multi-class response?
- Can generalize the logistic function to n targets?
- One vs All

One Vs All (or Rest)

- Separate classification problems
- One class variable vs Not-that-Class
- Do this for all target variables



(Reference: Simple Linear Regression: Step-By-Step - Dan Wellisch)

One Vs All (or Rest)

- Each classifier gives its probability of its target class
- We take MAX among them

One-vs-all

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$.

On a new input x, to make a prediction, pick the class i that maximizes

$$\max_i h_\theta^{(i)}(x)$$

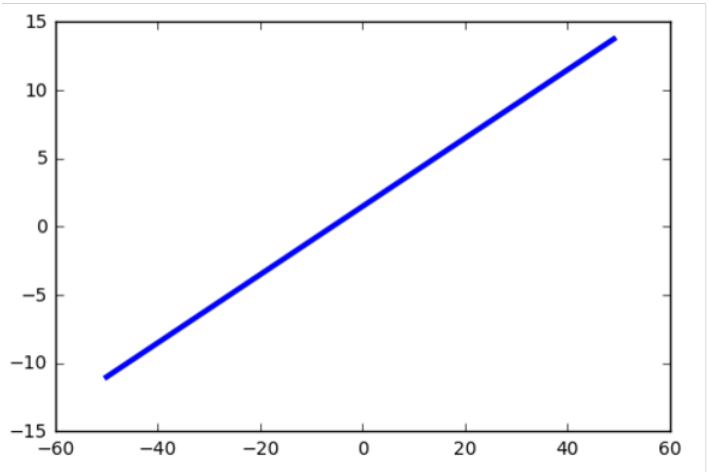
Summary

Regression vs. Classification

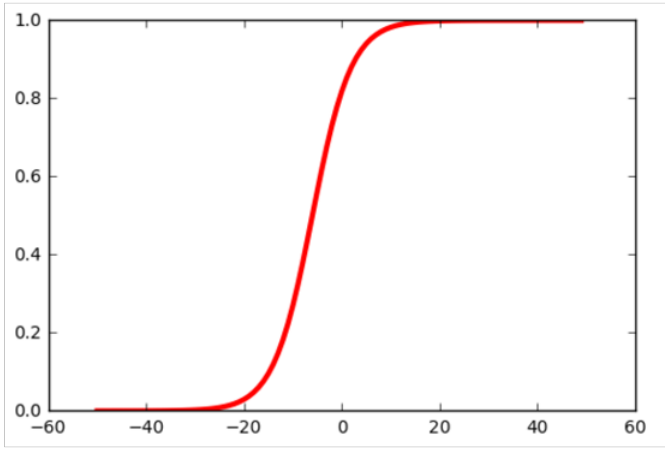
- In standard linear regression, the response is a continuous variable (float). It can take any value, large/small, positive/negative, integer/float, etc.
- In logistic regression, the response is qualitative/discrete (int, enum)
- Binary responses: Success or failure, Spam or Ham (Not Spam), etc They are bound to only categorical values such as 0,1.
- Its like converting analog signal (values, any float number) to digital signal (0 and 1s only)
- Note: Independent variables (Xs) can be of any type ie continuous or discrete, for both, Regression and Classification problems. They are differentiated based on the type of the target (y) only. If target is continuous then its a Regression problem and if the target is discrete then its a Classification problem.

Linear Model vs. Logistic Model

$$p(X) = \beta_0 + \beta_1 X$$



$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$



Logistic Function: outputs between 0 and 1 for all values of X

Logistic Regression with Scikit-Learn

Logistic Regression

```
# Logistic Regression
from sklearn import datasets
from sklearn import metrics
from sklearn.linear_model import LogisticRegression
# load the iris datasets
dataset = datasets.load_iris()
# fit a logistic regression model to the data
model = LogisticRegression()
model.fit(dataset.data, dataset.target)
print(model)
# make predictions
expected = dataset.target
predicted = model.predict(dataset.data)
# summarize the fit of the model
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
```

(Ref: Machine Learning Algorithm Recipes in scikit-learn - Jason Brownlee)

Thanks . . .

- Feel free to follow me at:
  - Github (github.com/yogeshhk) for open-sourced Data Science training material, etc.
  - Kaggle (www.kaggle.com/yogeshkulkarni) for Data Science datasets and notebooks.
  - Medium (yogeshharibhaukulkarni.medium.com) and also my Publications:
    - Desi Stack https://medium.com/desi-stack
    - TL;DR,W,L https://medium.com/tl-dr-w-l
- Office Hours: Saturdays, 2 to 5pm (IST); Free-Open to all; email for appointment.
- Email: yogeshkulkarni at yahoo dot com