Friday, 4 July 2025

Machine learning notes

Cross validation

* Since there are many machine learning models, we might be confused which model to choose, here comes the cross valdation
* There are two steps involved
  + we need to estimate parameters for the model which is also known as training the data
  + Next we need to check how well the model performs
* Suppose we have a large block of data we can use 75% of data for training the model and the rest 25% of data to train the data. But we dont know which 75% of data to take.
* Cross validation takes each 75% data once and estimate the parameters and test the model on the rest 25%. and gives an average proformance of the model.
* If we divide the data into 4 parts it is called 4 fold cross validation

Confusion matrix

* A confusion matrix is a matrix that tells us what our algorithm did right and what it did wrong
* Example is a model that predicts whether a person has a heart disease or not. So a confusion matrix is such that the rows indicate the prdicted decision and colums represent true decison

|  | Person has heart disease | Person does not have hear disease |
| --- | --- | --- |
| Person has heart disease | 1 | FALSE POSITIVES |
| Person does not have heart disease | FALSE NEGATIVES | 1 |

Sensitivity and specificity

* After making the confusion matrix, we can find the sensitivity and specificity which are defined as follows
  + Sensitivity is defined as the percentage of true positives over true positivies and false negatives.
  + Specificity is defined as the percentage of true negatives over true negatives and fasle positives

Example

|  | Person has heart disease | Person does not have hear disease |
| --- | --- | --- |
| Person has heart disease | 139 | 20 |
| Person does not have heart disease | 32 | 112 |

Sensitivity calculation:

True positives = 139

false negatives = 32

therefore sensitivity = 139/(139 + 32) = 81.3%

Specificity calculation:

True negatives = 112

false positives = 20

therefore specificity = 112/(112 + 20) = 85%

Example for 3 x 3 confusion matrix

|  | Movie A | Movie B | Movie C |
| --- | --- | --- | --- |
| Movie A | 12 | 102 | 93 |
| Movie B | 112 | 23 | 77 |
| Movie C | 83 | 92 | 17 |

Sensitivity for movie A:

True positives = 12

False negatives = 112 + 83 = 195

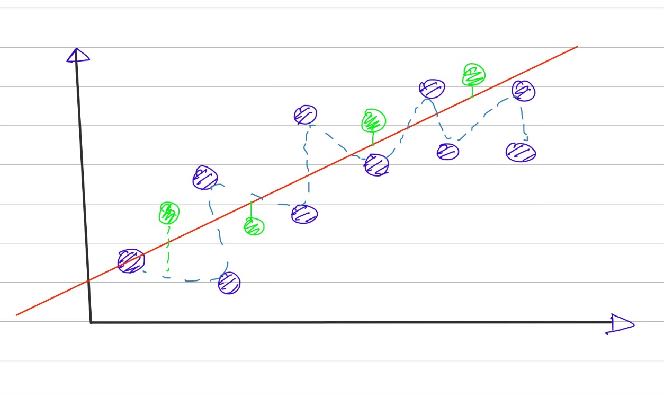
sensitivity(movie A) = 12/(12 + 195) = 6%

specificity for movie A:  
True negatives = 23 + 77 + 92 + 17 = 209

False positives = 102 + 93 = 195

specificity (movie A) = 209/(209 + 195) = 51.73%

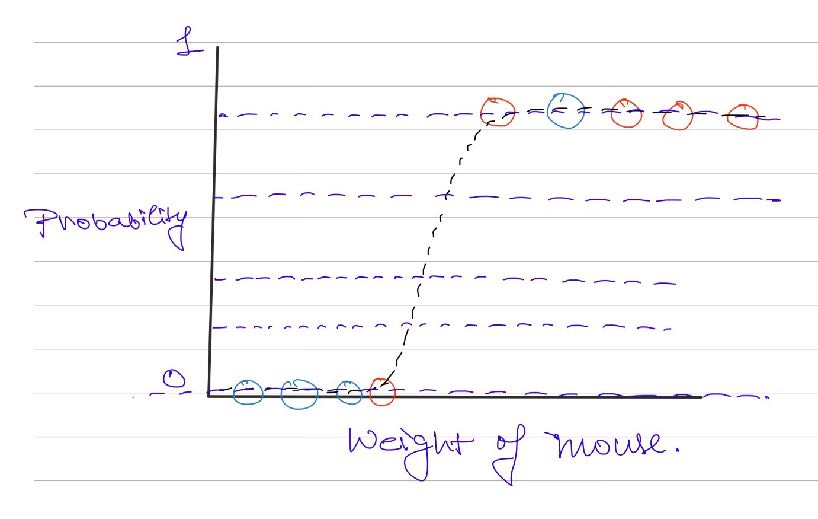
Bias and variance

* Variance is defined as how the model fits the datasets , example:

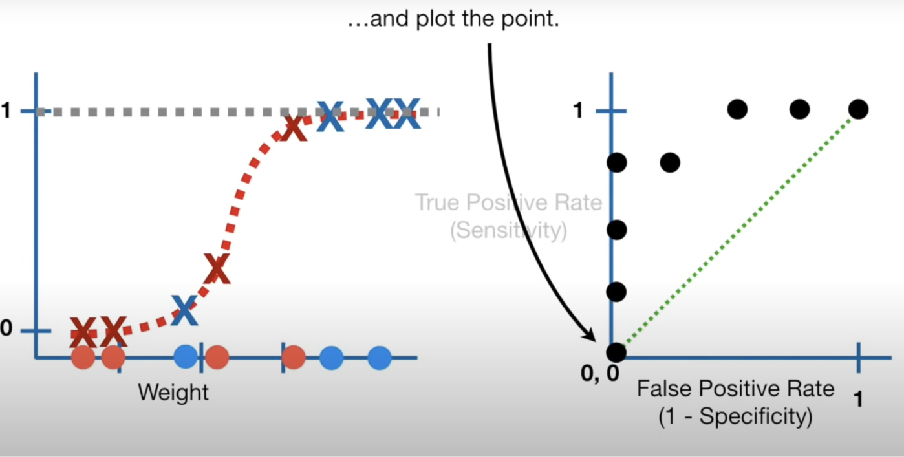
The squigly line has more variability than the straight line

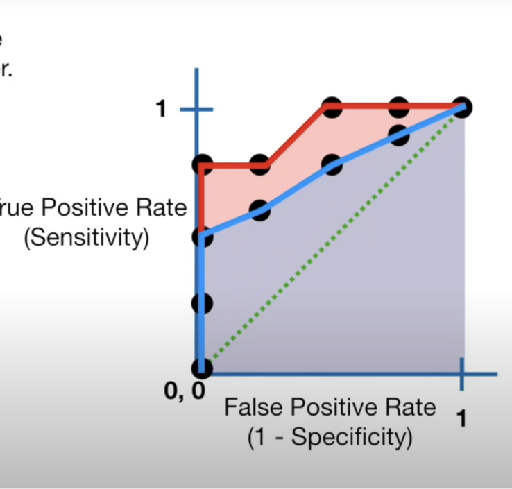
* Bias: It is defined as how well the model fits the real world data. The squigly line has less bias than the straight line.

ROC AND AUC

* Suppose we want to classifiy subjects on basis of some of its characterstics like height, weight etc
* For eg, we want to classify mice as obese and non obese on the basis of their weights

As you can see we need to set the threshold so that we can classify the mice

Each threshold value creates a confusion matrix .These large number of confusion matrices can be confusing so we plot a graph called ROC(reciever operator characterstic) which plots rate of false positives (1 - specificity) vs rate of true positives (sensitivity)

The area under the ROC gives AUC (area under the curve) . The larger the AUC the better the classification model

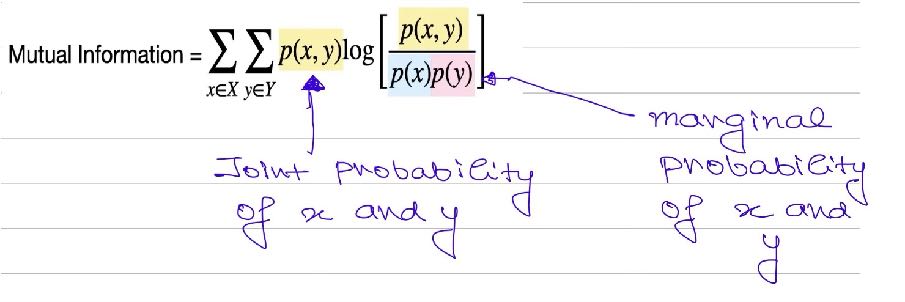
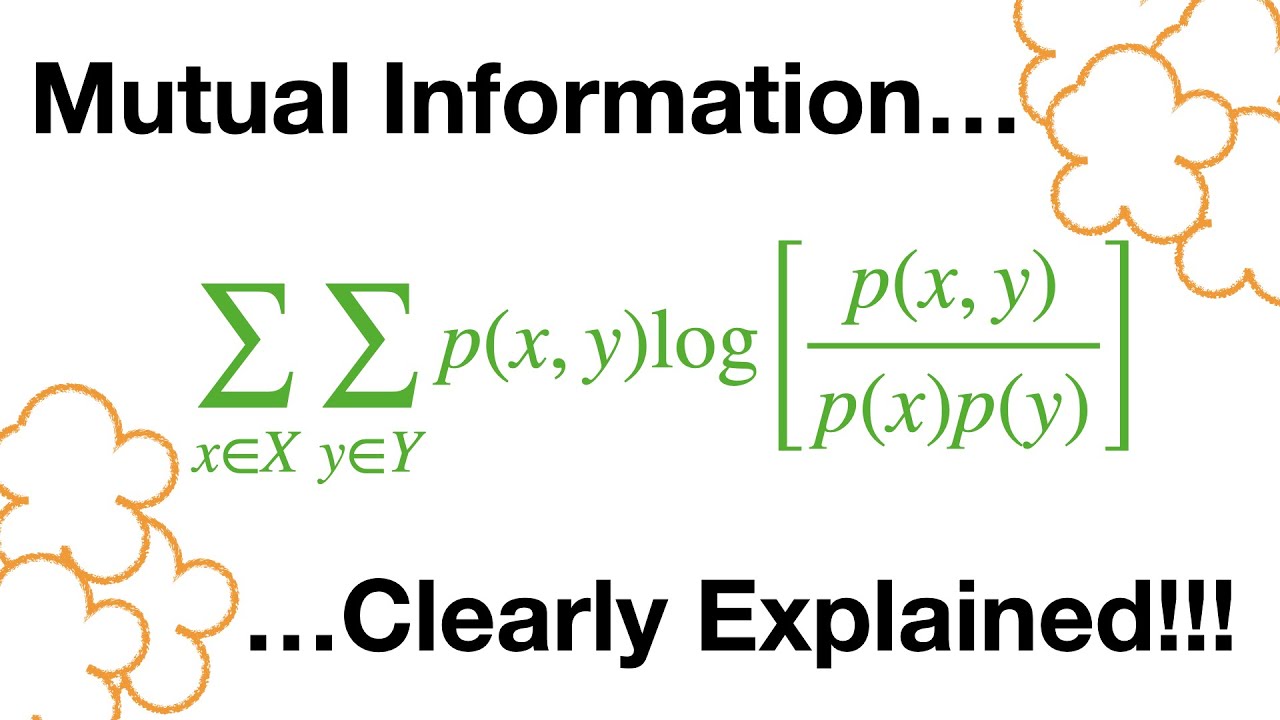
Red is AUC for logistic regression and blue is for random forest

Thus logisitic regression is better than random forest method

Entropy

* Before learning about entropy , we need to know about surprise
* Surprise can be seen as some sort of inverse of probability
* Instead of directly taking inverse of probability, we take logarithm of inverse of probability
* We define entropy as expected value of surprise
* suppose we have a coin with probability of head as 0.9 and probability of tails as 0.1, and we flip the coins 100 coins, we estimate the surprise of heads as (0.9x100)xs(head) and surprise of tails as (0.1x100)xs(tail). Therefore total surprise of 100 flips is estimated as (0.9x100)xs(head) + (0.1x100)xs(tail).
* Then we divide this estimated value by number of flips and thats how we get entropy. Thus we can write entropy as

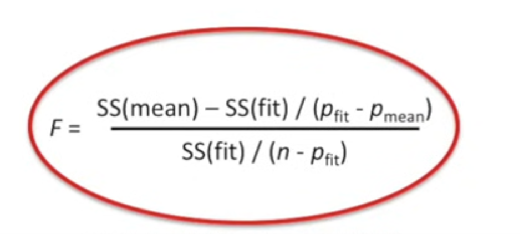
Mutual Information

* Mutual information tells us about how closely two variables are .
* Mutual information for discrete variables is given by
* Refer to the video below[](https://www.youtube.com/watch?v=eJIp_mgVLwE)
* For continuous variables , we can make a histogram and use the bins as the dicrete values
* The number of bins is equal to number of rows and columns for calculating mutual information

Fitting the best line

* While fitting a line to the observed data , the main concept is to minimize the square of distance of observed data from the data
* This can be done by plotting the sum of squares with respect to parameters and find the derivative where it is zero, the value of the parameters where the derivative of sum of squares with respect to parameters is zero is the best fitted line
* Here the parameters refer to the slope and intercept of the fitting line

Linear Regression

* Linear regression quantifies the relationship between two variables
* The parameter which quantifies how good the relationship is R-square value which is defined as var(mean) - var(fit)/var(mean)
* Variance around mean is calculated by taking square of difference between observed value of y and mean value of the observed values of y and taking the sum of these squares and dividing by the number of points
* Variance around fit is calculated by taking square of distance between the observed value and the fitted line and summing the squares and dividing by the number of observations
* A large R-squared value means a good fit
* But to tell if the calculated R-squared is statistically significant, we calculate the p value which is calculated from the F distribution
* F is given by

Where rho\_fit is number of paramters in fitted line and rho\_mean means number of paramters in mean line

We first generate random points and caluclate the f values we do this a large number of times and create a histogram of the f values obtained

The p value is the number of more extreme values divided by the total number of values.

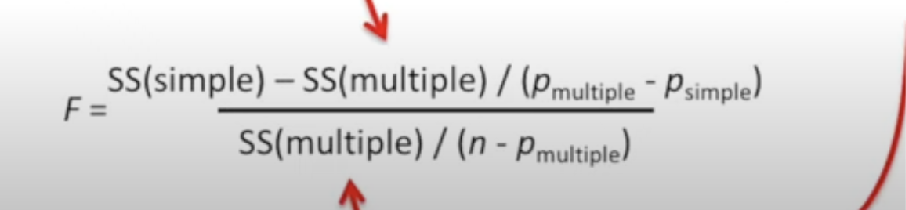
The p values are smaller when there are more samplesrelative to the number of paramters in the fit.

We want this p value to be small.

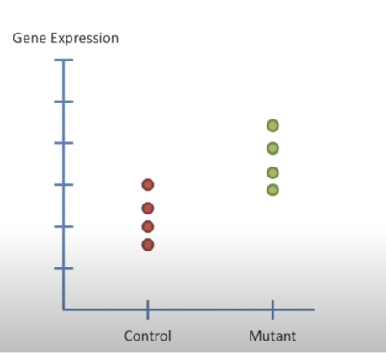
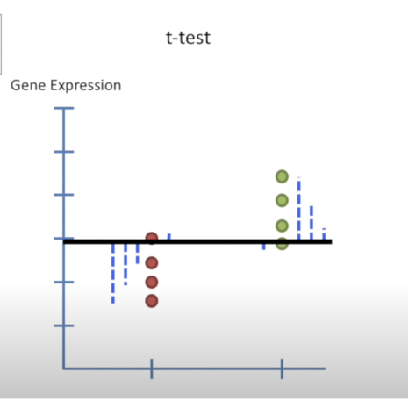
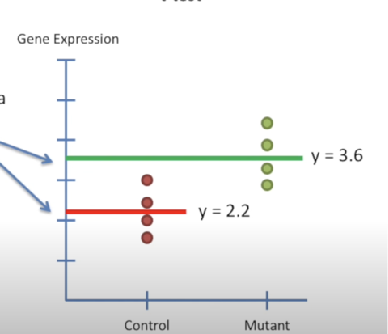
In linear regression , we want to know whether an independent variable contributes to the dependent variable or not, so our null hypothesis is beta\_1(coefficient of independent variable we need to check) = 0 and we got beta\_1 to be 250 from fitting the best line, p value is defined as if our null hypothesis is true, how probable is the event that the parameter takes extremes value,

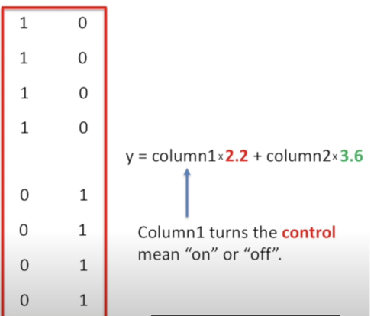
* If we calculate the p value using the values , we get p value as 0.002 which means there is 0.2 percent chance that we get extreme value , thus we can reject null hypothesis

Multiple linear regression

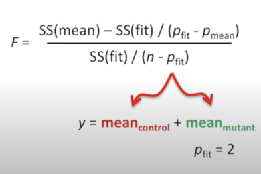
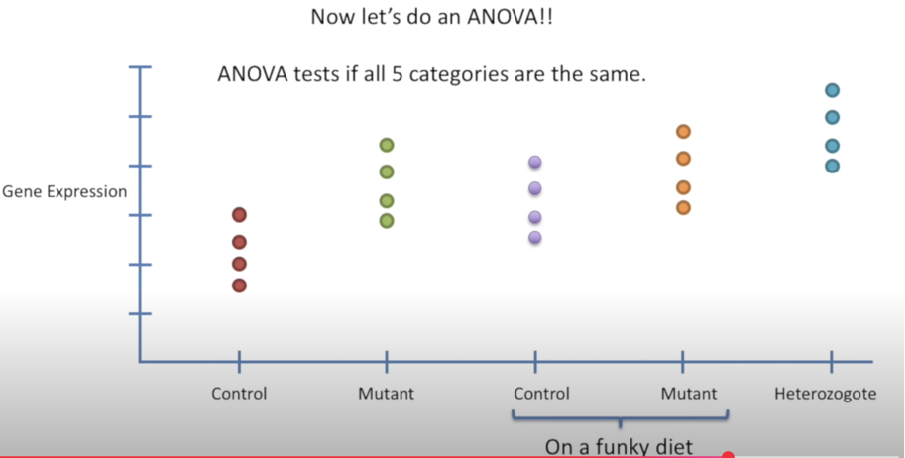
* The formula for calculating the R-squared values and p values for multiple regression is same as that of simple linear regression.
* The main thing about the multiple linear regression is that we can compare it with simple linear regression to tell whether the addition variable was worth taking or not.
* The formulas for calculating the R-square and p value are similiar, where the only difference is that instead of mean line we have simple linear regressed line and in place of simple linear regressed line we have multiple regressed line
* If the r squared value is large and p value is small then it is benefeicial to consider the additional variable

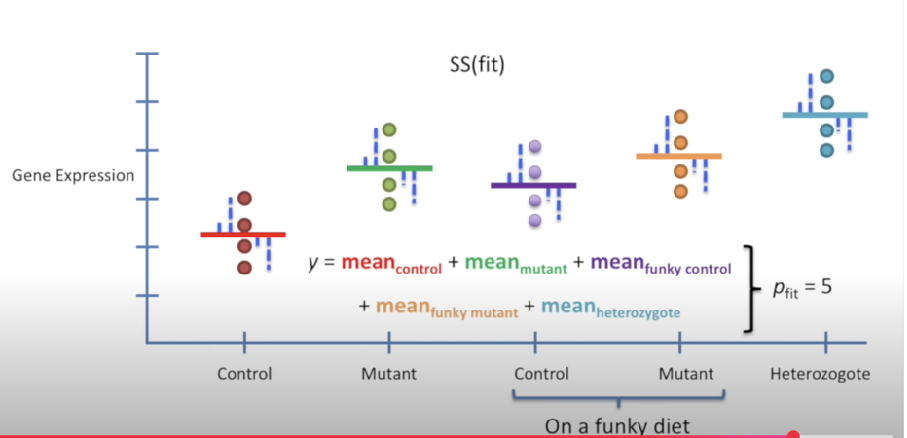
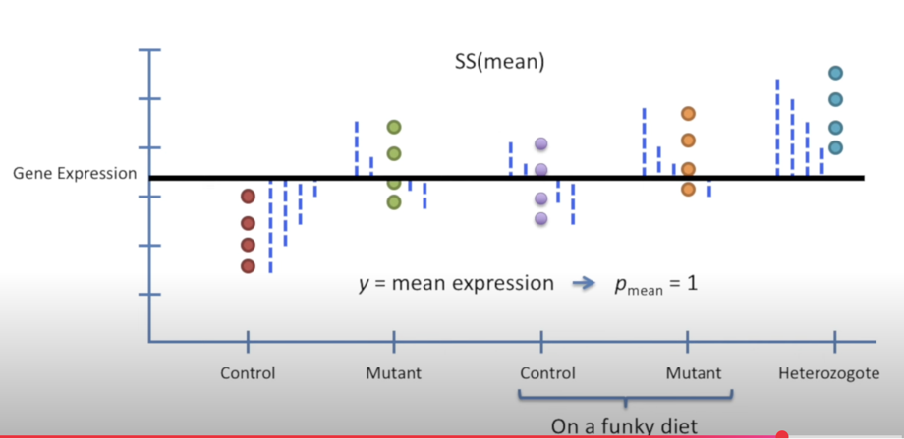
t-Test and ANOVA

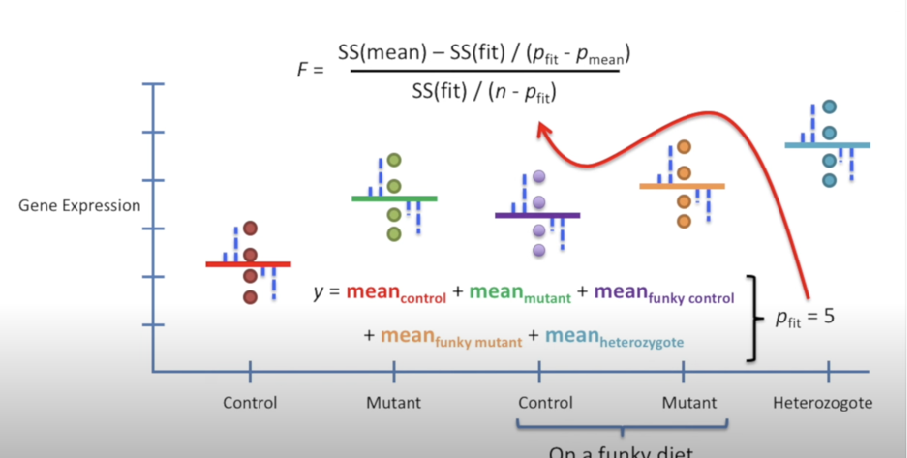
* t-Test is used for testing whether two means of two populations are same or not, this done by calculating the p values similiar to how we calculated for the linear regression
* Suppose we are given data of gene expression of mice with control gene and mice with mutant gene.
* First we calculate the overall mean and sum of square of residuals around mean
* Now we want to fit a line to the data, the least squared fit is the mean of control mice and mean of mutant mice
* Now we want to combine the two lines, for a point of control mice we can write it asImage

The residual is the distance of point from the least squared fitted line which the mean of control mice, similiarly we can write the equation of each point and we get

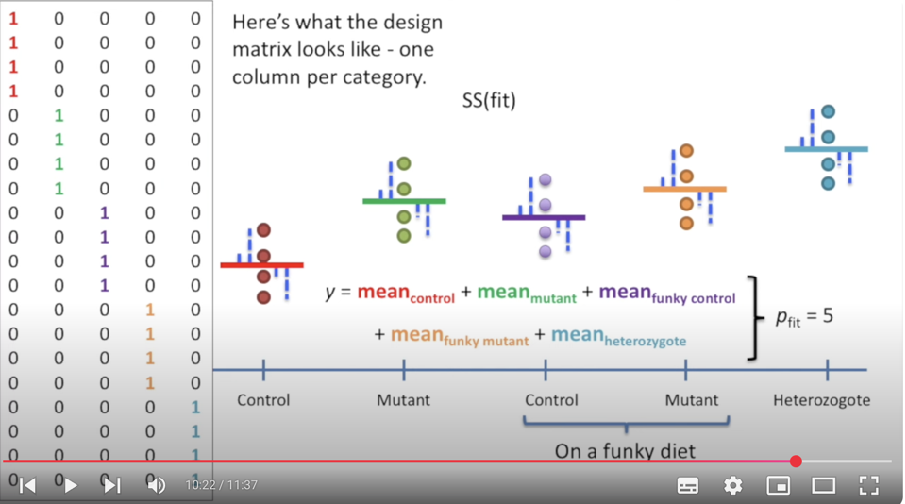
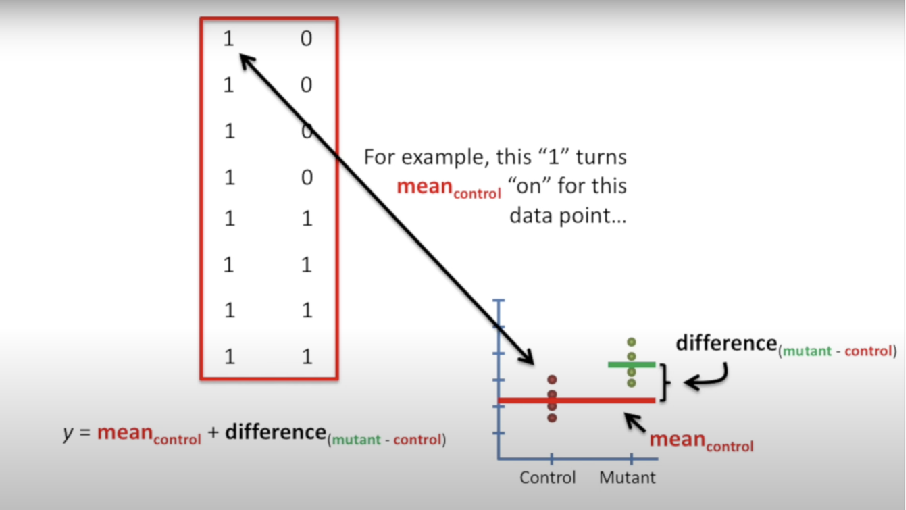
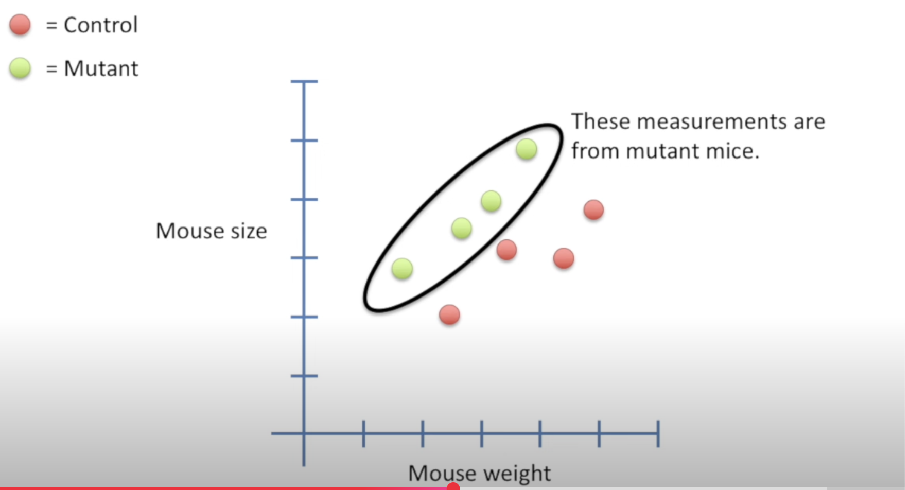
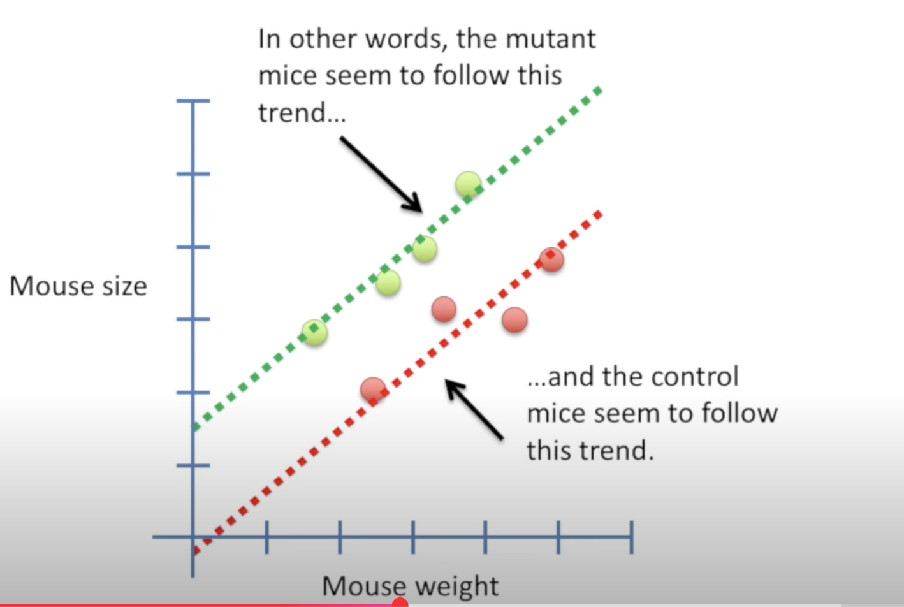
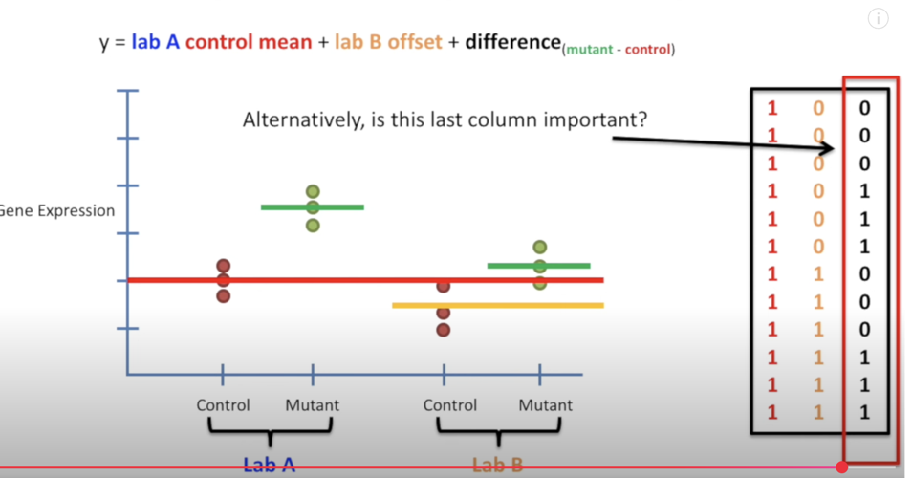
Image

* The matrix is called design matrix
* Now we can use the formula for F used in linear regression to calculate F value in above case, here rho\_fit = 2 (mean\_control and mean\_mutant). Thats how we calculate the F value for t test
* ANOVA tests if all the categories are the same for eg .





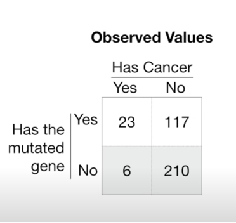
More about design matrices

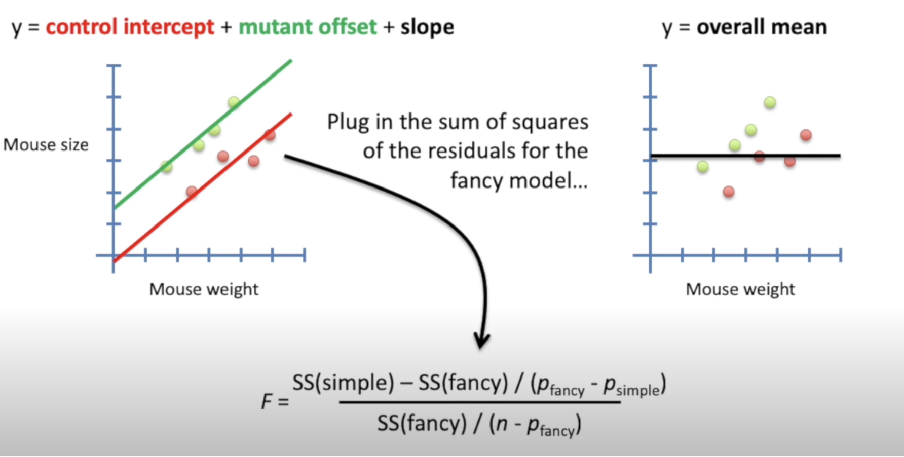
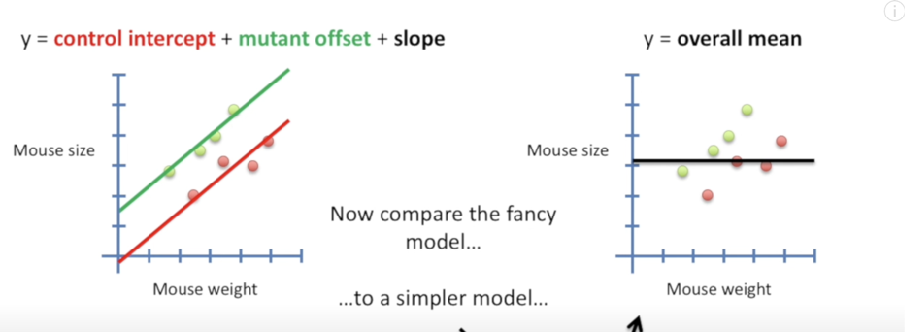
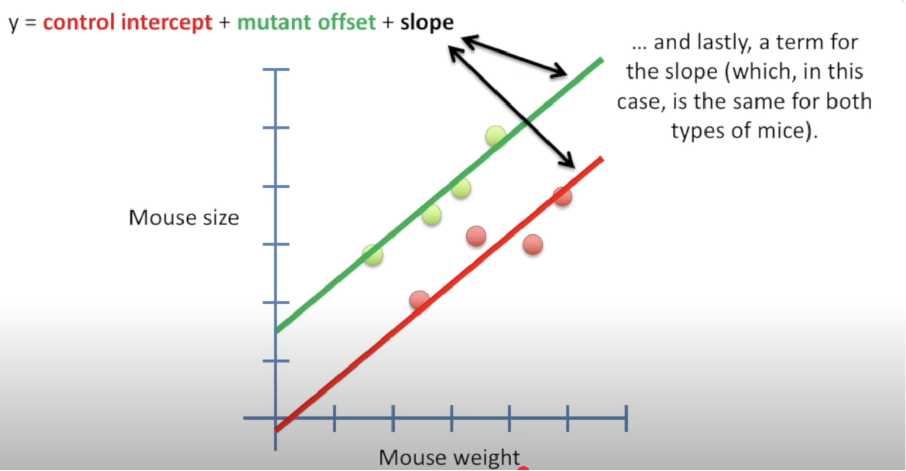
* Design matrices above are not commonly used , the ones commonly used are as follows
* It takes into account the offset between mutant mean and control mean
* Now we combine t-test and linear regression
* We want to compare the two lines
* Rho\_fancy would be 3 since there are three paramters (control intercept, mutant offset and slope)

Odds and log of Odds

* Odds refer to ratio of something happening to something not happening
* For eg, “Odds of my team winning is in favour 6 to 1” means odds are 6/1 = 6
* To make things symmetrical , we generally calculate log of odds
* One can also find odds from probability which is log(p/(1-p)) where p is the probability of something happening
* The log of odds is also called logits and is used in logistic regression

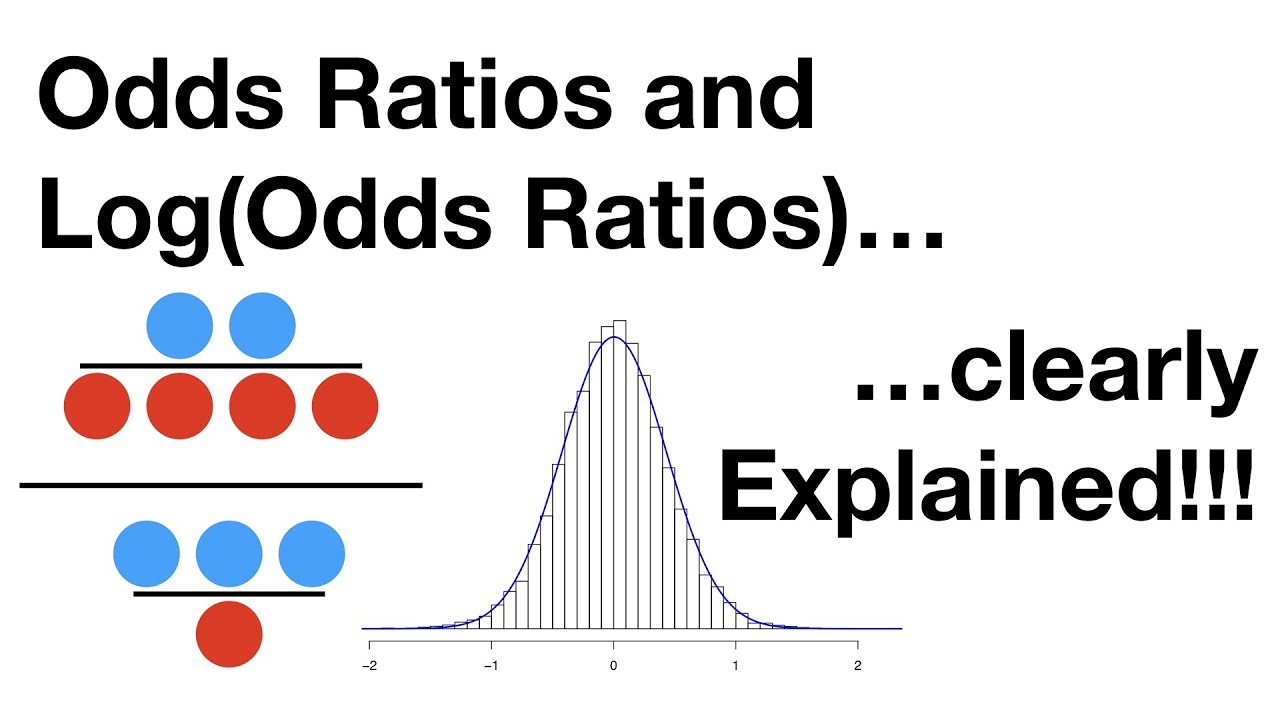
Odds ratio and log of odds ratio

* Odds ratio is basically ratio of two odds
* For eg, we want to know whether having mutated gene results in higher chance of having cancer, so we calculate odds ratio which is 

In numerator, given mutated genes , odds of getting cancer = 23/117 = 0.196

In denominator, odds of getting cancer without having the mutated gene = 6/210 = 0.02857

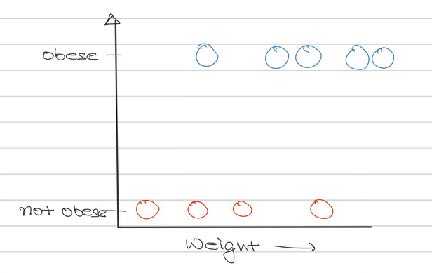
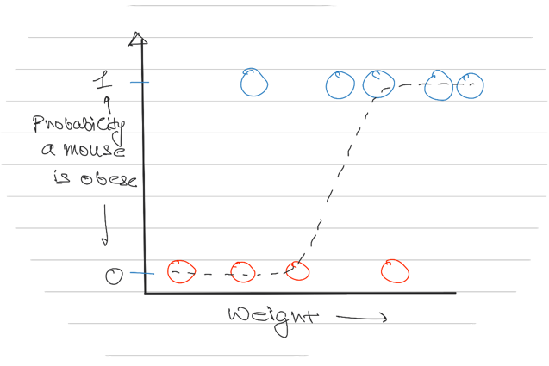
Therefore the odds ratio is 6.8803

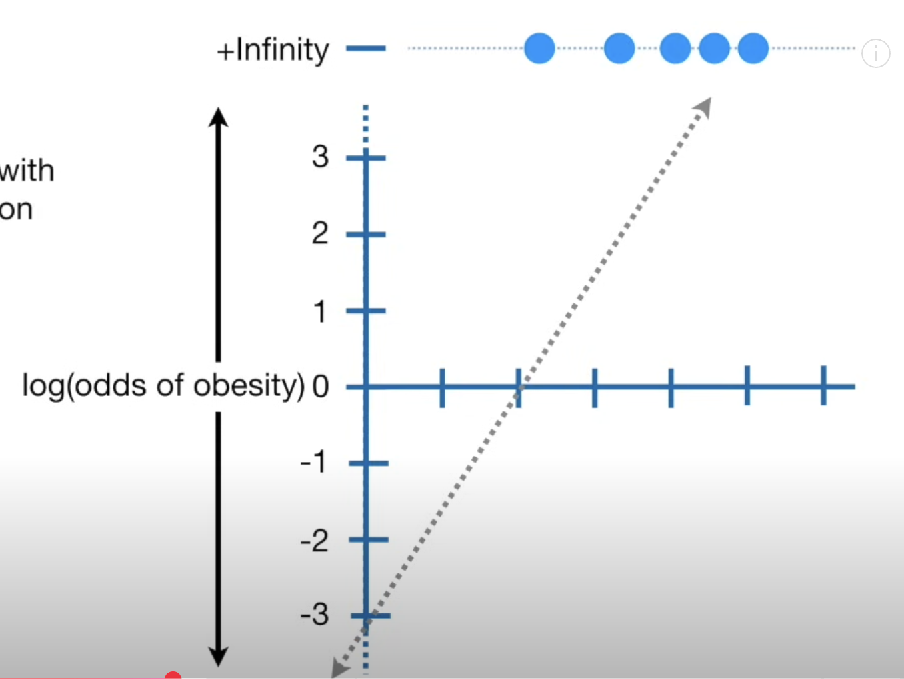
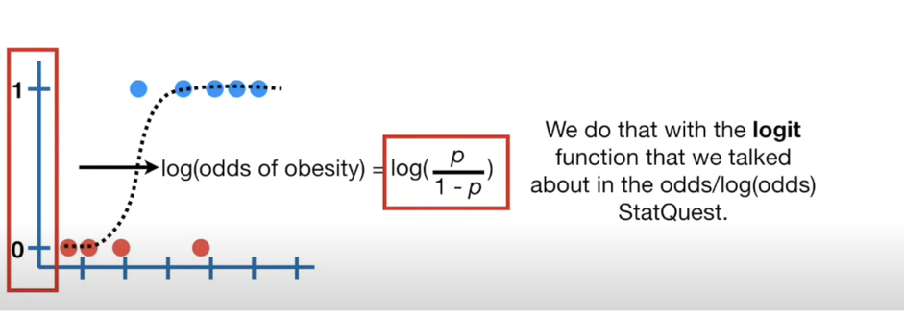
* We can see odds ratio as a similiar prameter like R-squared value.
* Next we want to check if it is statistically significant which is done by calculating the p value.
* There are three methods to calculate the p value which are
  + Fischer exact test
  + Chi- squared test
  + Wald test
* Wald test is different from the other two in the way that it calculates the p value and also gives the confidence interval.
* For more refer to the video below[](https://www.youtube.com/watch?v=8nm0G-1uJzA)

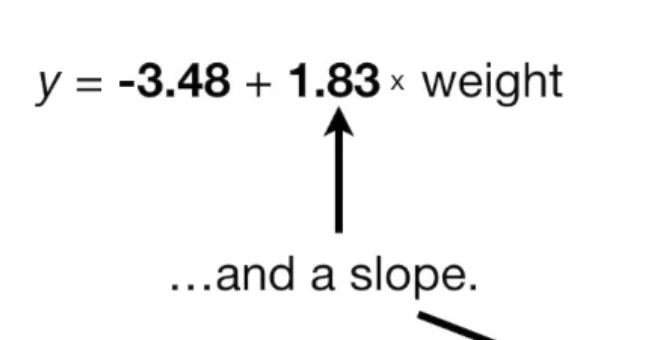
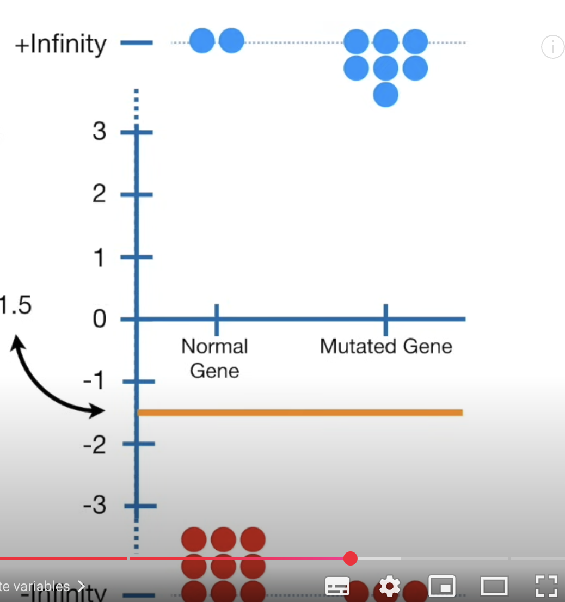
Logistic regression - Introduction

* Logistic regression is used for classification. It fits a logistic curve (S-shaped curve to the data) which gives probability
* Unlike linear regression, logistic regression is done by taking the curve which maximizes the likelihood.

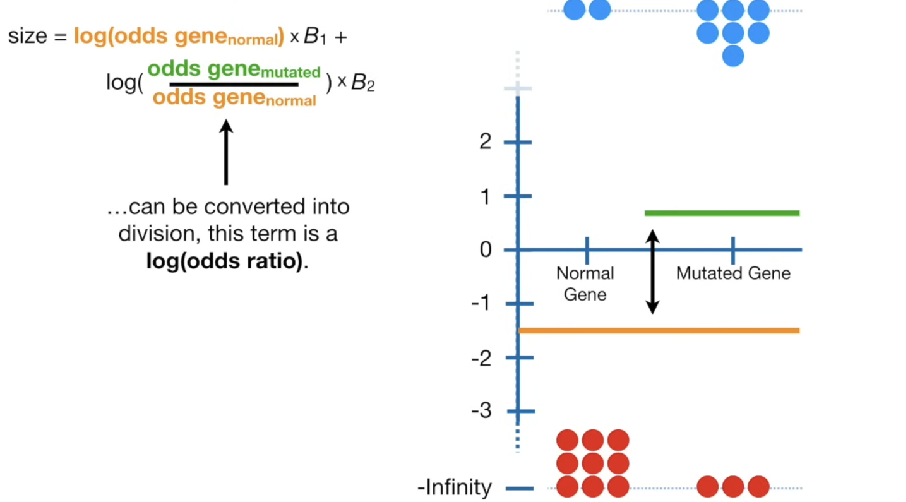
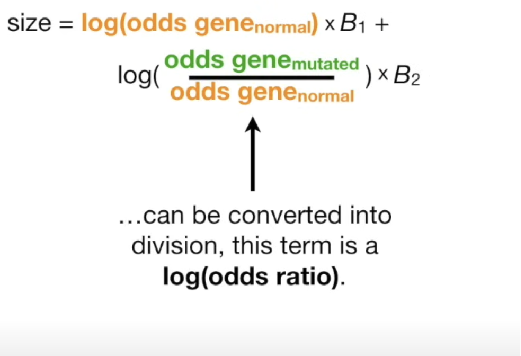
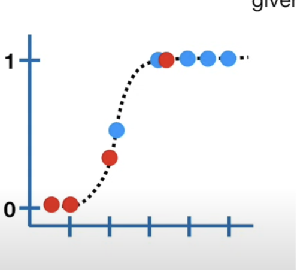
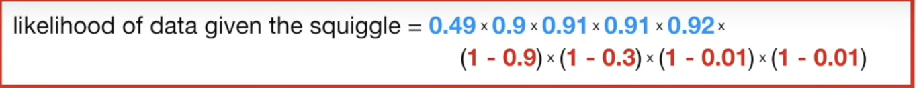
Logistic regression - 1

* Lets understand logistic regression using an example, we take weight of mice on x axis and categories obese or not obese on y axis
* Now we fit a logistic curve to the data
* Logistic regression is a part of GLM (generalized linear model)
* To make it more closely related to linear regression model , the y axis is transformed to log of odds of obesity

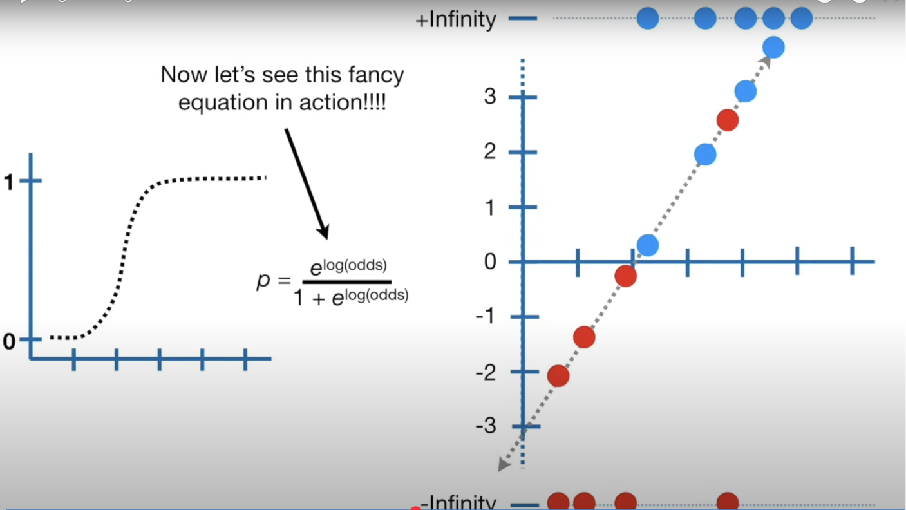
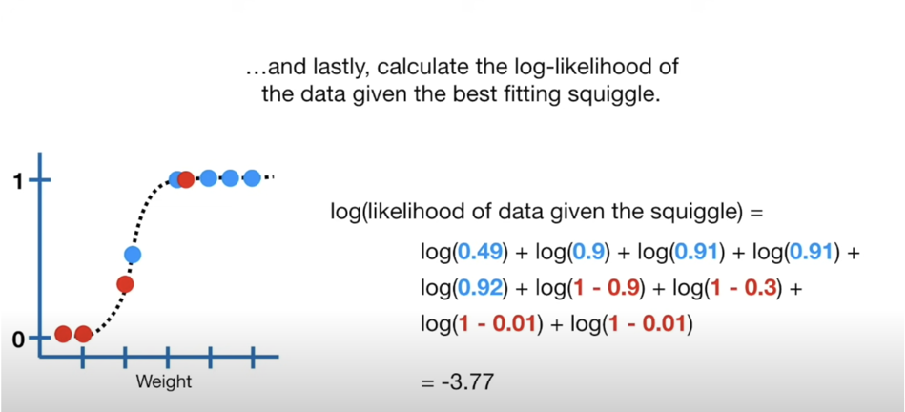
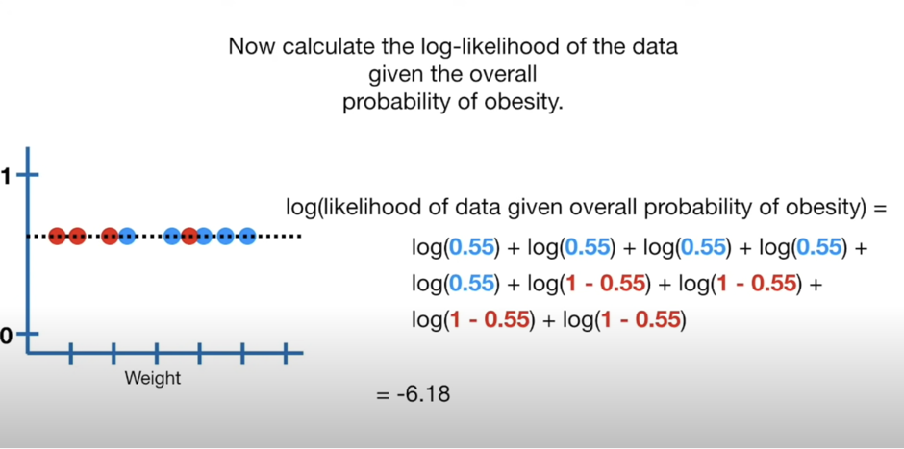
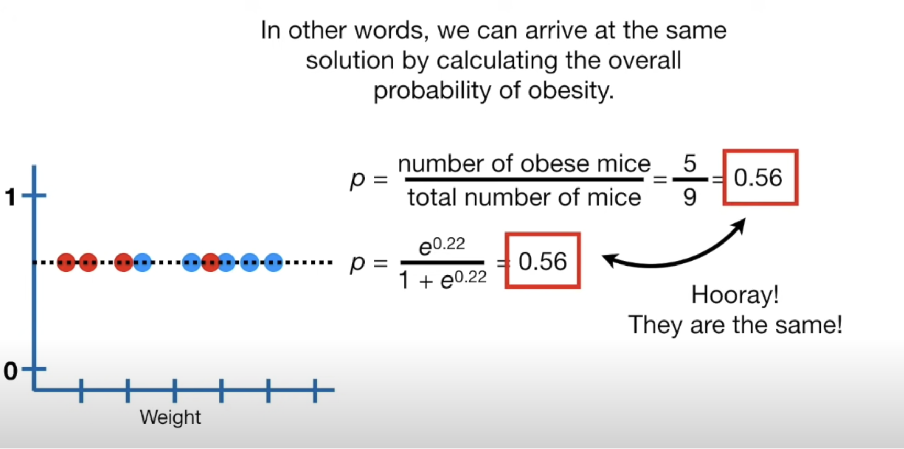
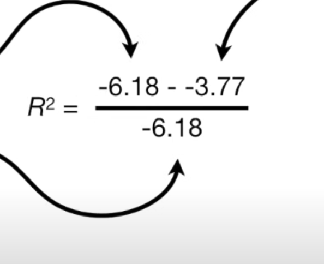


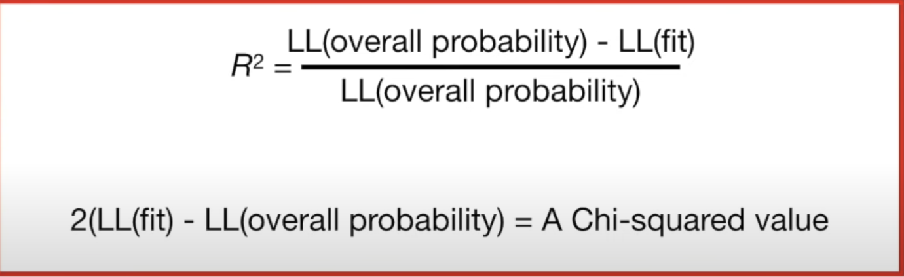
* The slope of the line is given as.
* We can also find the coefficients for discrete variables, unlike in the previous case when the weight was a continous variable we can have dicrete variables too
* We convert the y axis to log of odds of being obese and calculate the log of odds for each discrete variable
* Similiar to linear model we can perform t test for the discrete variable case

Logistic regression - 2 : Fitting the curve - maximum likelihood

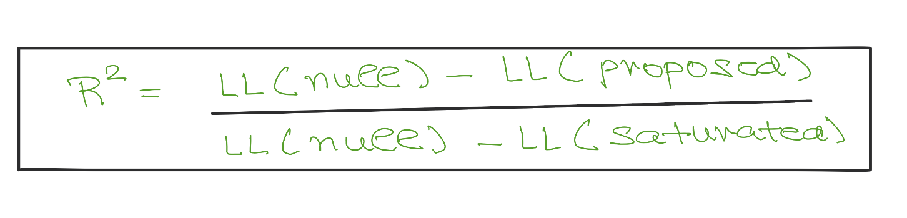
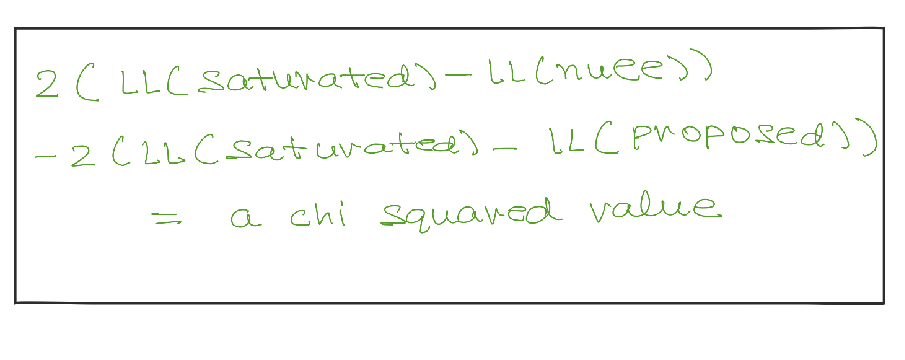
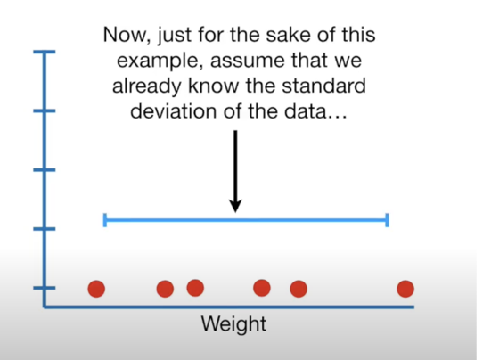
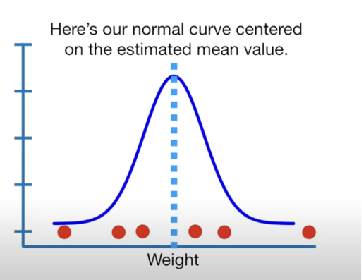
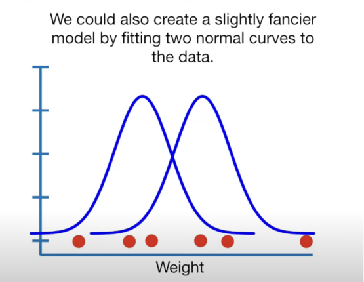
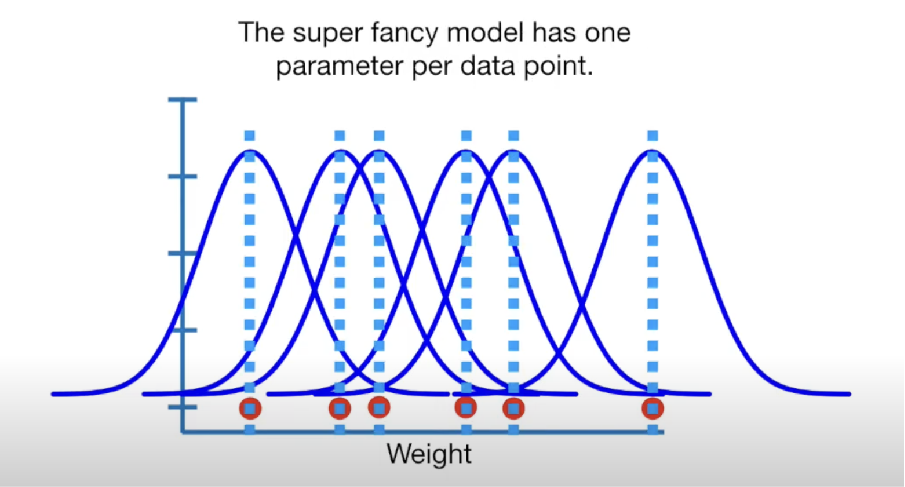
* To fit a logistic curve to data, the process is different from that of linear regression
* First we choose a candidate line on log of odds curve , then we project the data on the line and calculate the log of odds of that point, then we calculate the probability for that point
* Next we figure out the likelihood of mice that are obese given the squigly line and likelihood of mice that are not obese given the squigly lines and multiply them
* For red dots (mice who are not obese) , we calculate the likehood of not being obese as 1 - likehood of being obese
* The line which gives the maximum value of the likelihood is the best fitted line

Logistic regression - 3: R-squared and p values

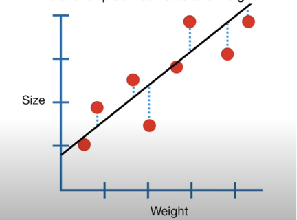
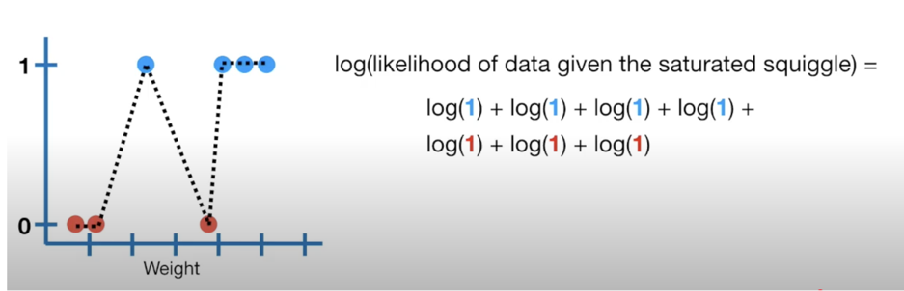
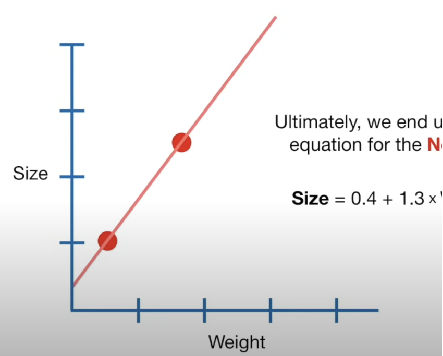
* There are many methods to calculate R-squared values, one of the method which is similiar to linear regression is given below
* In the formula for R-squared for linear regression instead of ss(mean) and ss(fit) we have ll(fit) and ll(mean) where ll is log likelihood
* To calculate ll(fit) we need to first project the data points on the best fit line for log of odds graph and then calculate log likelihood for each data point , that is the ll(fit)
* Next we calculate the overall probability of obesity
* LL(overall probability) = -6.18
* Therefore R-squared is given as
* To calculate p value, the method is straight forward, we find the a chi squareed value which is given by :

2(LL(fit) - LL(overall prob)) where the degrees of freedom of chi squared value is given by difference in parameters of fit and overall prob which is 2(slope and intercept) - 1(intercept) = 1

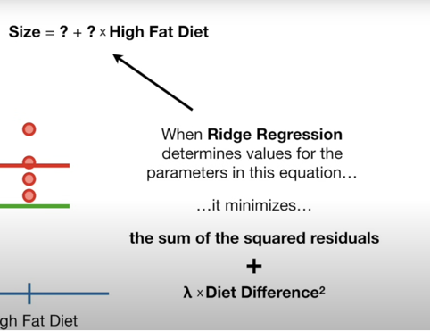
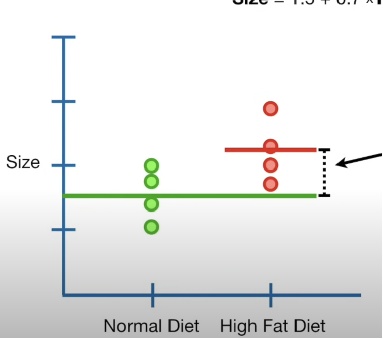
Saturated model and residual deviances and null deviances

* The expressions for R-squared value and p value are not commonly used , the one commonly used is
* The expression for R-squared is called Mc Fadden R-squared value.
* The chi squared value has dof equal to parameters in proposed model - parameters in null model
* To explain saturated model, lets take an example , suppose we have some weights and we know the standard deviation of the data, we estimate the mean of the data and fit a normal curve (model) to the data
* Since we had to estimate only mean, we say the model is 1 parameter model
* The log likelihood of the data points given the normal model is -3.51
* We can also create a fancier model by fitting two normal curves to the data
* Since we need to estimate only the two means (we still assume that we know the standard deviations) , we say the model has 2 parameters
* The log likelihood of this model is 1.27
* The above model which has 1 parameter for each data point is called saturated model and it gives the maximum log likelihood for the data.
* The log likelihood of the data points given the model is 7.16.
* The null model gives a sense of worst case scenario and the saturated gives a sense of best case scenario
* We want the likelihood of the proposed model to be greater than null model and close to the saturated model
* The saturated model provides an upper bound for the best fit.
* The log likelihood of saturated model ensures that the R-squared value remains between 0 and 1.
* Lets talk about the deviance:
  + Residual deviance is defined as : 2(LL(Saturated model) - LL(Proposed Model))
  + This gives a chi squared value with degrees of freedom as parameters in saturated model paramters in proposed model)
  + Similiarly null deviance is defined as : 2(LL(Saturated model) - LL(Null Model))
  + This gives a chi squared value with degrees of freedom as parameters in saturated model paramters in Null model)
* The p value of the R-squared value is given by Null deviance - Residual deviance.
* Now we talk about why we ignore the LL(saturated model) while calculating R-squared value and p value, in saturated model , the squiggle fits the data perfectly so the log likelihood would be 0

Ridge Regression - 1

* Lets take an example to understand ridge regression, suppose we have data points for mice as their sizes and weights
* Least square line accurately reflects the relationship between Size and Weight
* Suppose we are given only two points then the least square fit line would give
* Suppose we are also given the test data
* As we can see the red line has a high variance , in machine learning lingo , we say that the red line over fits the training data
* The idea behind ridge regression is that we introduce a small amount of bias to the new line in order to get a significant drop in the variance
* The least square tries to estimate the parameters in the equation :
  + Size = y-intercept + slope\*Weight

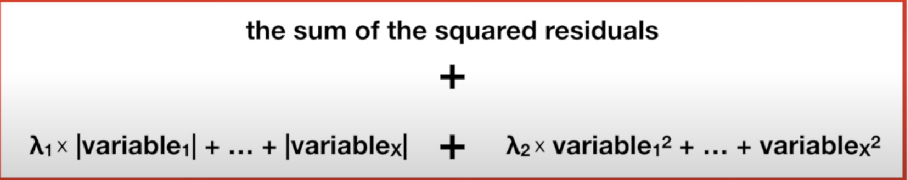
By minimizing the sum of square of residuals

* In contrast to this ridge regression minimizes the sum of square of residuals + lambda\*(square of the slope), this methods adds a penalty to the least square method and lambda determines how severe the penalty is.
* As we increase the value of lambda , the slope of the ridge regression line decreases which means Size becomes less sensitive to small changes in weight
* The value of lambda is determined using cross validation, typically 10 fold cross validation, it is a positive value
* Ridge regression can also be used for discrete variable like:
* The whole point of doing ridge regression is because small sample sizes like this can make poor predictions
* In logistic regression, the ridge regression minimizes the sum of likelihoods + lambda \* (slope)^2
* Suppose we have a large number of parameters in our equation, so ridge regression minimizes the sum of square of residuals + lambda\*(all parameters^2 except y-intercept)
* Normal least square regression requires n data points to determine n paramters, but if we have lesser number of data points than the parameters, we can use ridge regression to determine all n parameters using those data points only

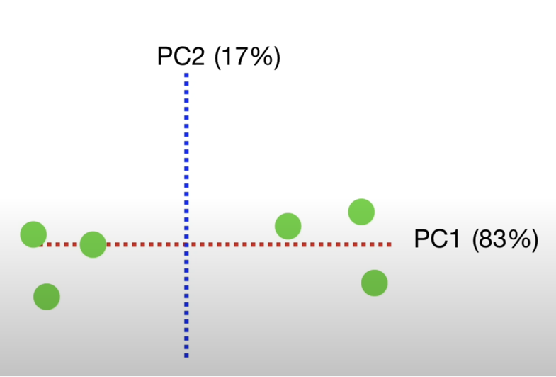
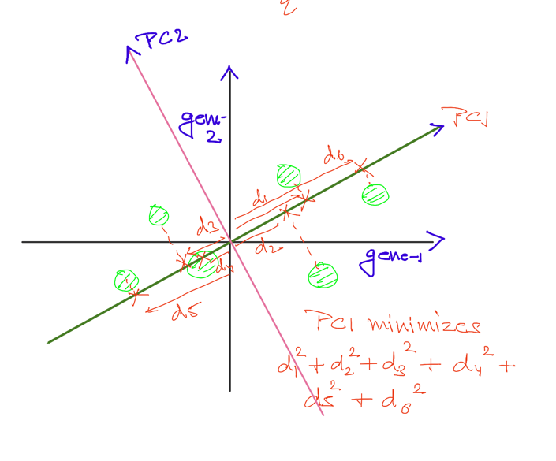
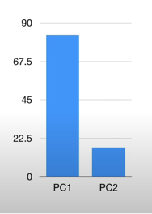
Lasso Regression

* Lasso regression is very similiar to the ridge regression, the superficial difference between them is that in ridge regression, the penalty is parameter^2 , where as in lasso regression the penalty is |paramter|.
* The major difference between ridge and lasso regression is that despite the large value of lambad , in ridge regression, the slope can never be zero it can only assymptotically to be zero but in lasso regression, the slope can be zero.
* Suppose we are given a en equation with a large number of variables where some of them are useless , lasso regression can make the parameters associated with these useless variables to be zero
* Therefore lasso regression works best when our model has large number of useless variables

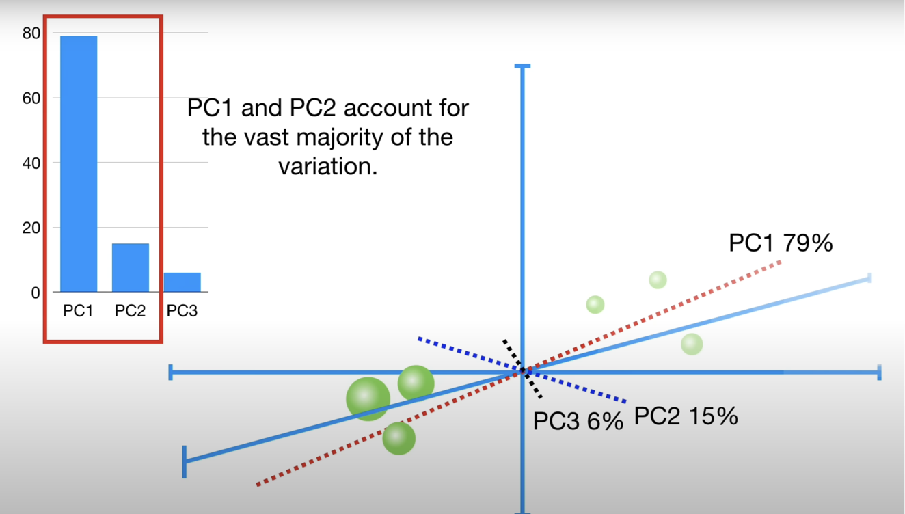
Elastic net regression

* Suppose we have tons of variable in our model and we dont know whether they are useful or useless in advance
* Instead of using lasso or ridge regression we can use elastic net regression
* The elastic net regression works well when there is correlation between parameters

Principal Component Analysis

* Let’s start with an example:
* Suppose we are given only gene 1 as variable so we can plot the data on a single line and tell which mice cluster together
* Then if we are given gene 2 and gene3 , we can plot it on 2d and 3d graph respectively and tell which mice cluster together
* But if we are given 4 or more genes , then it becomes difficult to plot , here PCA comes which can take 4 or more genes and can make a 2D pca plot.
* PCA also tells which pca variable decided the clustering
* Steps to do to PCA:
  + Lets start with simple case when we are given only 2 genes, we plot the data on a 2D plot and find the average value of y components and x components and center the data at the average point
  + Next find a line which minimizes the sum of square of perpendicular distance of points on the line or maximizes the distance of projected points on the line.
  + This line is called PC1 (principal component 1).
  + To make PC1 , we have 4 parts gene 1 and 1 part gene 2, this is called linear combination of gene1 and gene 2,
  + When we do pca ,we find the unit vector in the direction of line, this is called eigen vector for PC1, also the sum of squares of projected points from the origin divided by n-1 is the eigen value of PC1.
  + PC2 is the line perpendicular to the PC1 line .
  + To make the PCA plot , we simply rotate the two lines such that PC1 is horizontal and usiing the projection of points on the lines, we plot the data onto this new 2D PCA plot.
  + Eigen values are simply the measures of variations for PCs, for eg PC1 accounts for 15/18 = 83% of the total variation around the PCs.
* Plot for percentage of variation contribution by each pca component is called SCREE PLOT.
* As we can see for case when 3 genes are given, we can see that PC1 and PC2 can explain 94% of the variation in data, thus a 2d pca PLOT would be sufficient

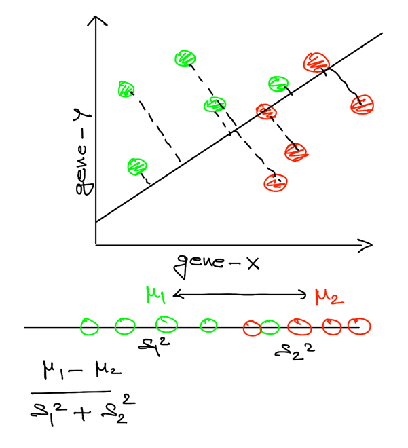
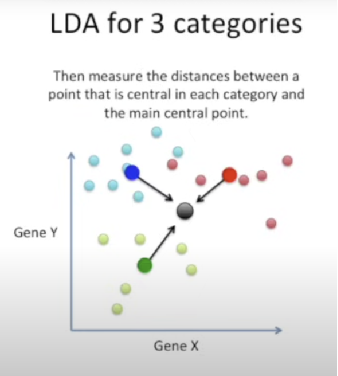
Practical tips for PCA

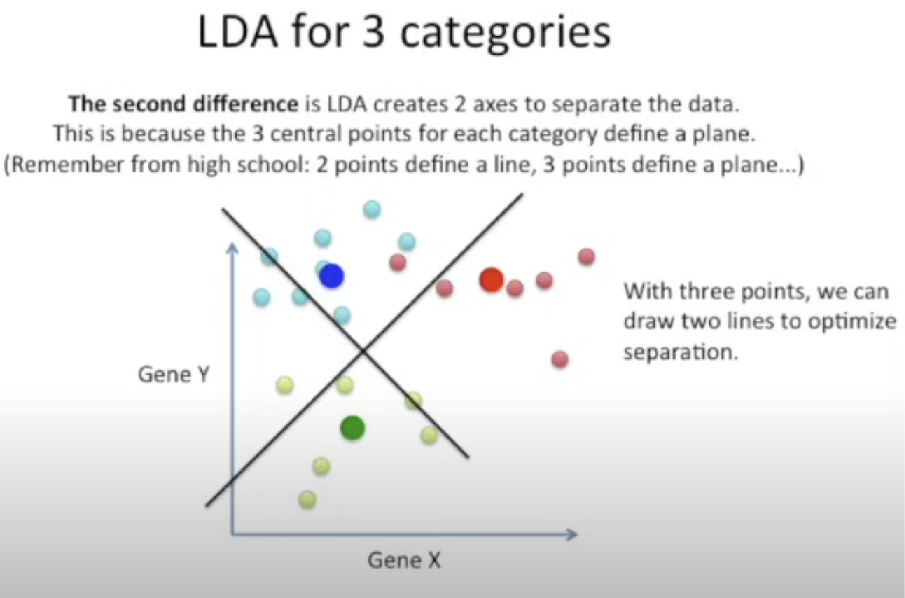
* Always scale your data, this prevents us from being biased to one of the variable, this is done by dividing each variable with its standard deviation
* Make sure the program you use for PCA does the centering
* Generally we have each PC for each variable , but if we have fewer number of samples than the variables , it puts an upper bound for the number of PCs we can expect.

Finding the PCs

* First find the mean vector for the sample data and subtract it from the samples.
* Then find the sample covariance matrix and then find the eigen values and eigen vectors related to this sample covariance matrix, order the eigen values from higest to lowest .
* The eigen vector with largest eigen value is the PC1 which explains most of the variation in the data.

Linear Discriminant Analysis

* LDA(linear discriminant analysis) is used when we are given data from various categories and we need to discriminate a new observation into one of the many categories we have, for eg , suppose we have some cancer patients on which the drug works and some on which it doesn’t and we want to find whether a new cancer patient belong to first category or the second category.
* LDA does this by creating an axis and projecting the data on that axis, the way LDA creates this axis is by maximizing the distance between the mean of the two categories while minimizing the scatter of each category around its mean.
* LDA is like PCA but instead of finding the genes which causes the most variation in the data , it focusses on maximizing the separatibility of the categories
* If more than two genes are given , the process is same
* If we have more than two categories , Find the overall mean and our goal is to maximize the distance of individual mean from the central mean
* The second difference is that LDA creates 2 axis in the order of importance how well the axis discriminates the data, the LD1 discriminates the most followed by LD2.

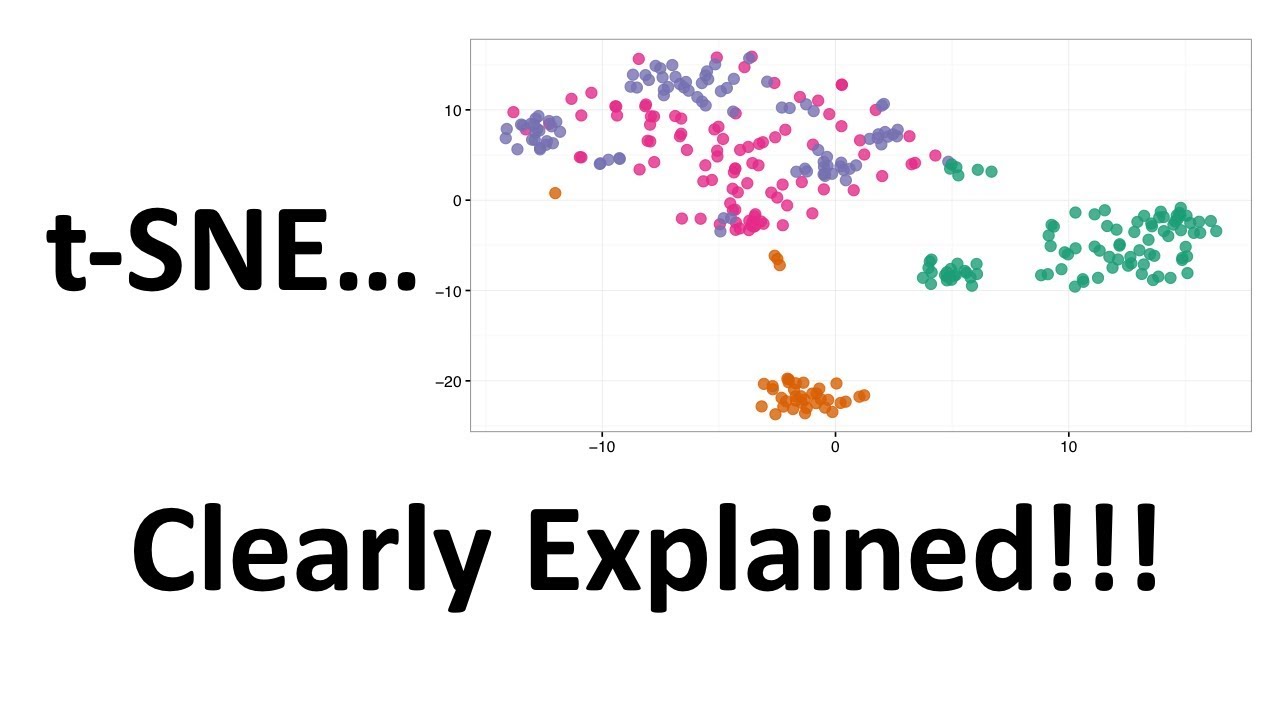


This helps us to reduce the number of dimensions, suppose we are given 10000 genes and it would be better if we can maximize the separation between the three categories

Multi dimensional Scaling and PCoA

* MDS is similiar to PCA except instead of using correlation , it uses distance between the data points, the closer the points are with respect to the distance , more closesly they are correlated.
* Suppose we are given a multi dimensional data , it becomes difficult to plot these data points, thus MDS finds a distance matrix of the data points and gives an x and y coordinate to each data point converting it to a 2D graph.
* In other words , clustering based on minimizing the linear distances is same as maximizing the linear correlations.
* The underlying maths and type of outcomes remain same for MDS and PCA.

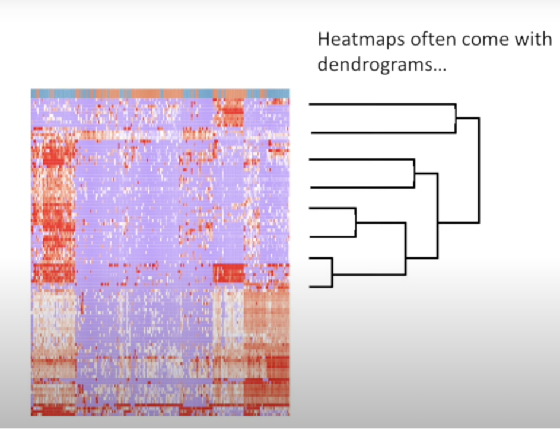
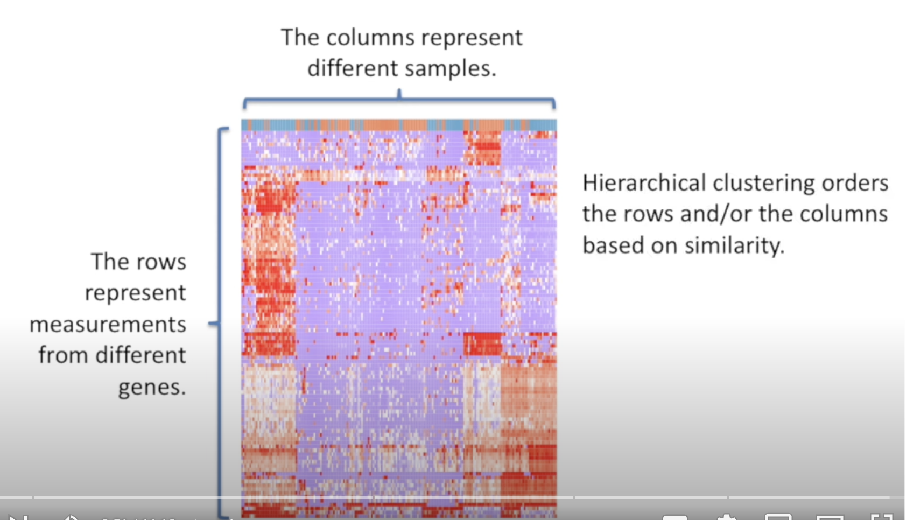
t-SNE

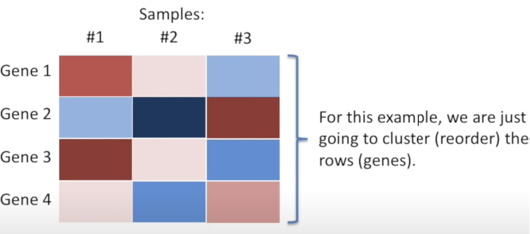
* Takes a high dimensional data set and converts into a 2 dimensional plot while retaining most of the information in the original dataset.
* The way it works that for a 2d case , it randomly plots points on a 1d line and moves each point at a time on the basis of how they are attracted and repelled to other points[](https://www.youtube.com/watch?v=NEaUSP4YerM)

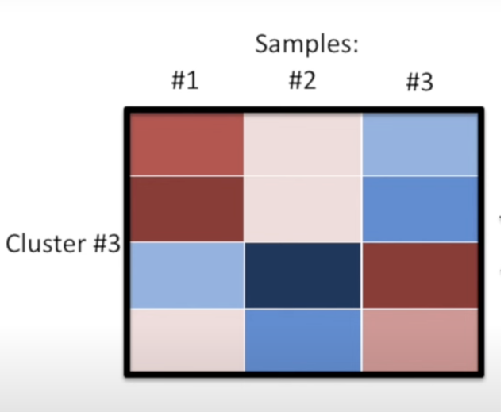
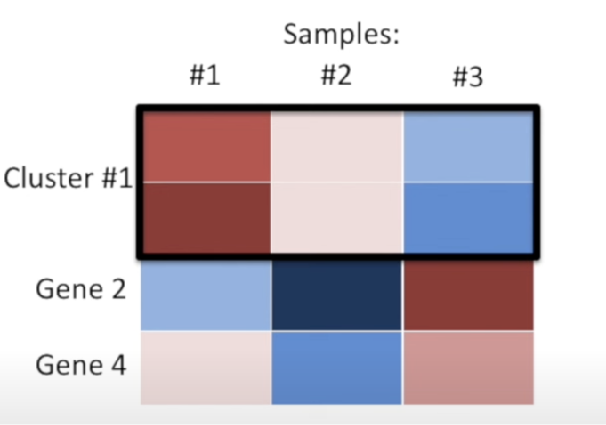
**How t-SNE works (simplified):**

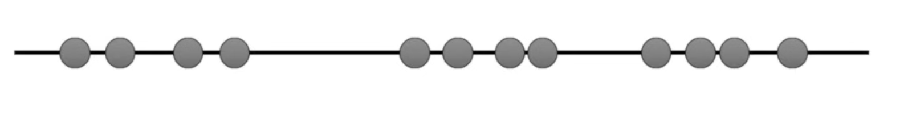
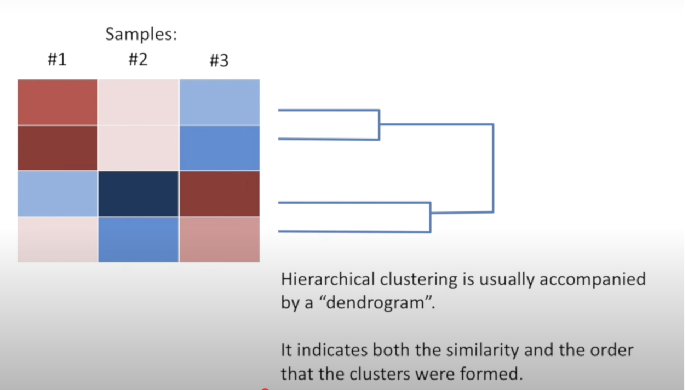
1. **Compute pairwise similarities** between all points in high-dimensional space.
2. Convert these similarities into **probabilities**—high for nearby points, low for distant ones.
3. Do the same in low-dimensional space.
4. **Minimize the divergence** between the two distributions using a cost function (Kullback-Leibler divergence).
5. Output the best 2D/3D representation that maintains neighborhood relationships

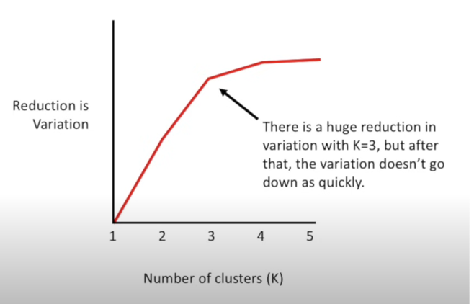
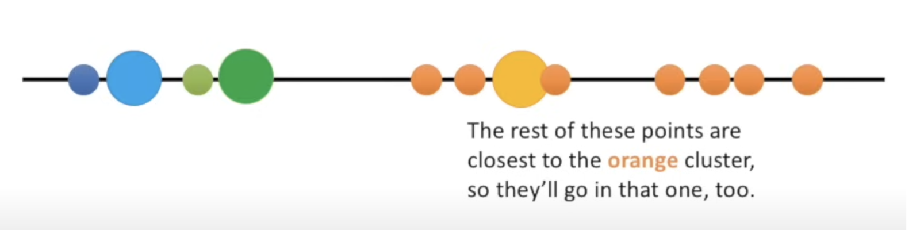
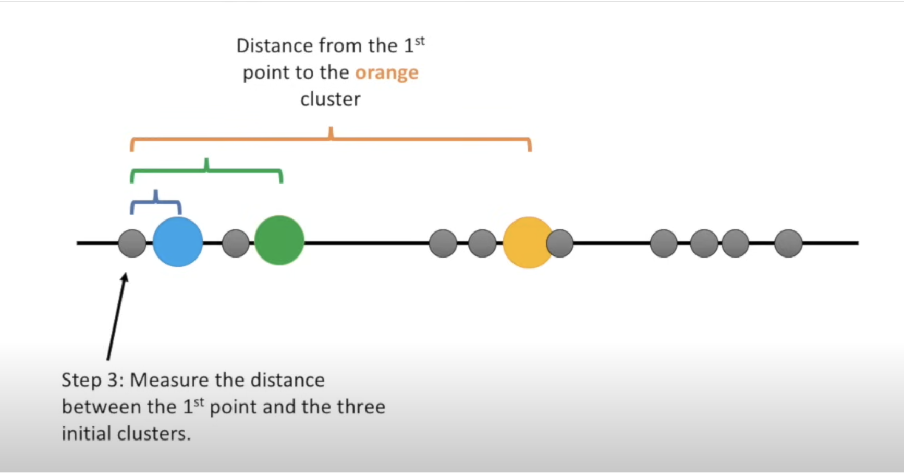
Heirarchical Clustering

* Heirarchical clustering is often associated with heat maps like
* It arranges the rows and/or columns based on some notion of similiarity



* To explain this lets take an example of 3 samples and 4 gene expressions
* Coceptually , in the first step find which gene expressions are similiar to gene expression 1 and do this for all other gene expressions,
* Finaly merge those two gene expressions which are most similiar
* Now treat the above formed cluster as a single gene expression and repeat step 1.
* Finally we get the following cluster
* The height of dendogram represents which gene expressions were most similiar , the one with lower heights were merged earlier means they were more similiar than others.
* To quantify similiarity , we use different metrics like euclidean distance , manhattan distance etc to create a distance matrix
* For comparing clusters, we can compare their average point (called centroid), or the closest point in each cluster (single linkage) or farthest point in each cluster (complete linkage).

K-means clustering

* K means clustering is done when we know how many clusters we want to have.
* Suppose we are given points on a number line
* Steps to perform k means clustering is
  + step-1: Choose randomly three points (here we have choosen k = 3 clusters),
  + step-2: Next we measure distance of each point from one of the intitial clusters and merge them in one to which they are most similiar.
  + step-3: find the mean of the three clusters formed and repeat the steps 1 and 2, if they do not change we have got our clusters
* K means clustering cant see the best clustering , it only keeps track if the total variations of the clusters formed and try to reduce it by choosing different starting points
* To choose best k, we plot the total variation vs k and when we get an elbow we choose that point to be optimal k
* For 2D case, we measure distance with respect to some metric like euclidean distance and the rest procedure is same

DBSCAN

**DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) is a powerful **unsupervised clustering algorithm** that groups data points based on **density** rather than distance (like k-means or hierarchical clustering).

🧠 **Core Intuition**

DBSCAN identifies clusters as **dense regions** of points separated by regions of **low density** (i.e., sparse data or noise).

📌 **Key Terms**

* **ε (epsilon)**: Radius of neighborhood around a point.
* **minPts**: Minimum number of points required to form a dense region.
* **Core Point**: Has at least minPts points (including itself) within ε.
* **Border Point**: Has fewer than minPts within ε but lies within ε of a core point.
* **Noise (Outlier)**: Neither a core nor a border point.

⚙️ **How DBSCAN Works**

1. Start from an unvisited point.
2. If it's a core point:
   * Create a new cluster.
   * Recursively add all density-reachable points (within ε and meeting minPts).
3. If it's a border point:
   * It’s added to the nearest core point’s cluster.
4. If it’s noise:
   * Mark as outlier.
5. Repeat until all points are visited.

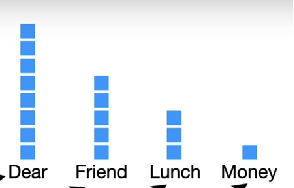
✅ **Advantages**

* **No need to specify number of clusters**
* Can find **arbitrarily shaped clusters**
* Robust to **noise and outliers**

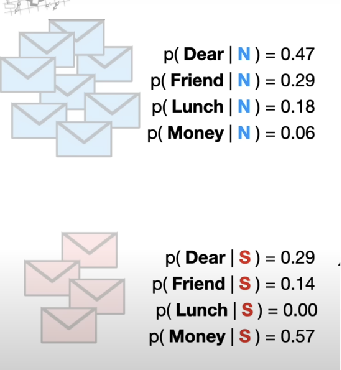
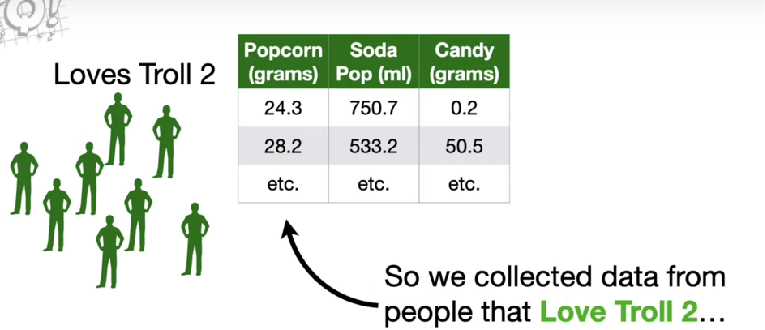
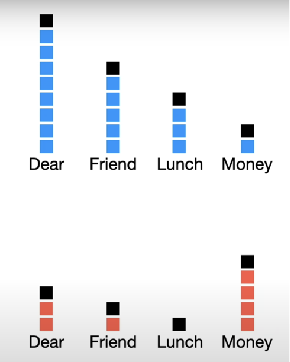
❌ **Disadvantages**

* Choosing good values for ε and minPts can be tricky
* Struggles with **varying density clusters**
* Performance can degrade with **high-dimensional data**

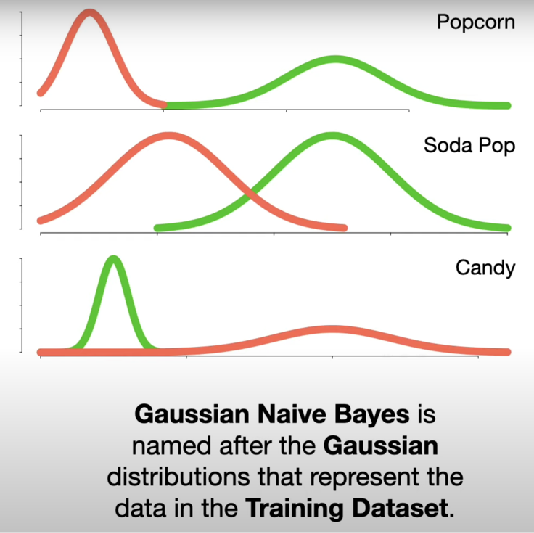
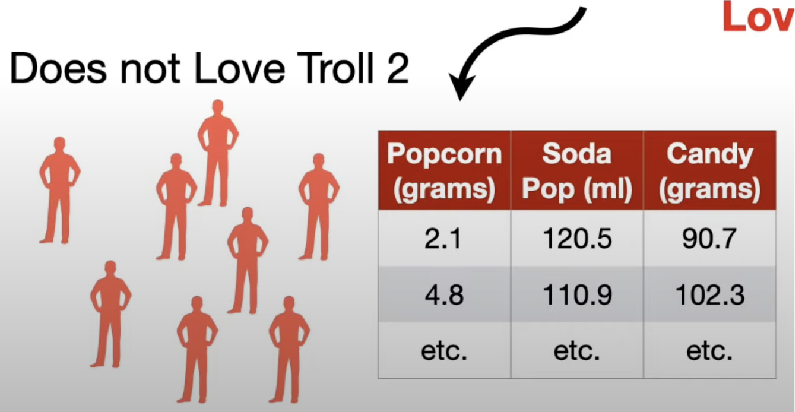
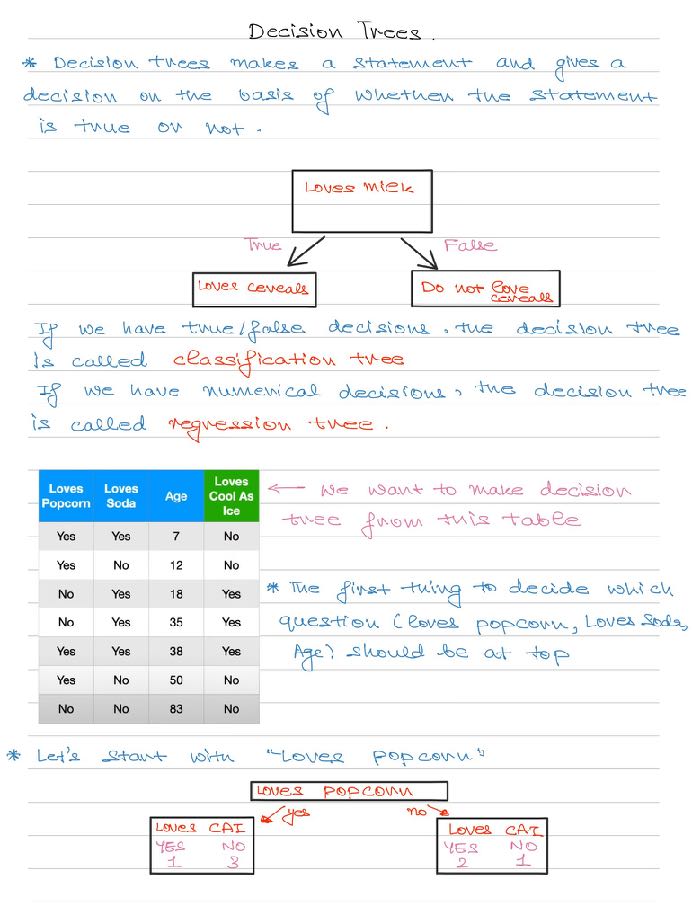
K-nearest neighbors alogorithm

* K-Nearest Neighbors (KNN) is a simple way to classify new data points based on proximity to previously labeled data. First, we label (or cluster) training data, potentially using dimensionality reduction like PCA for visualization. To classify a new, unlabeled point, we find its k nearest neighbors from the training data. The new point is assigned the label that is most common among those neighbors. In case of a tie, we may resolve it randomly or choose not to classify the point.

Naive Bayes

* Let us explain this using an example , suppose we recieved 8 messages from friends and family and 4 spam messages.
* We plot a histogram of words present in the messages
* Using the histogram we can calculate probability of each word given normal message like P(“Dear”|Normal) = 8/17 = 0.470
* Since we are calculating probabilities of discrete variables , these are also called likelihoods
* Now imagine , we get a new message “Dear Friend” and we want to classify it as spam or not spam.
* We start with an initial guess about the probability of any message being normal regardless of what it says which is P(N) = 8/12 = 0.67
* This initial guess is called prior probability
* Now we multiply the probability of dear that occurs in normal message and probability of friend that occurs in normal message. which is P(N) \* P(“Dear”|N) \* P(“Friend”|N) = 0.09
* However this is proportional to the probability that the message is normal given the message “Dear Friend”.
* Similiarly we do the calculation for spam case and we get 0.01. Thus the score for message being spam given the message is lower than the score for message being normal given the message, we classify the message as “Normal”.
* Lets take a more complicated example, lets say we recieve message “Lunch Money Money Money Money”.
* We might guess since the probability of word “Money” is more in spam messages the message would be spam.
* Lets do the calculation:
  + P(N|message) = P(N) \*P(Lunch|N) \*P(Money|N)^4 = 0.00002
  + P(S|message) = P(S) \*P(Lunch|S) \*P(Money|S)^4 = 0.0000
  + Since P(Lunch|S) is zero, to tackle this we generally add 1 count to each word in histogram
  + Now we get the value of P(N|message) = 0.00001 and P(S|message) = 0.00122 and we classify message as spam.
  + Naive Bayes is Naive because it treats all word orders equally, like it would treat “Dear Friend” same as “Friend Dear”.

Gaussian Naive Bayes

* Suppose we have people who love troll 2 and people who dont
* Next we draw gaussian distributionn curves for each column by calculating mean and standard variation of each column
* Now a person comes and eats 25 gm of popcorn , drinks 500 ml of soda pop and eats 25gm of candy every day.
* Now we take an initial guess of probability that the person loves troll 2 :
  + P(Loves Troll 2) = 0.5
  + P(Loves Troll 2| his eating habbits) propr to P(Loves) \* P(popconr = 20|loves) \* P(soda = 500ml | loves) \* P(candy = 25 | loves).
  + Taking log to prevent underflow, we get -124
  + Now we calculate the probabillity that the person does not love troll 2 , we get -48 > -124 and we classify the person as someone who does not loves troll 2.



