Conceptual Session
FDS, DAV & ML

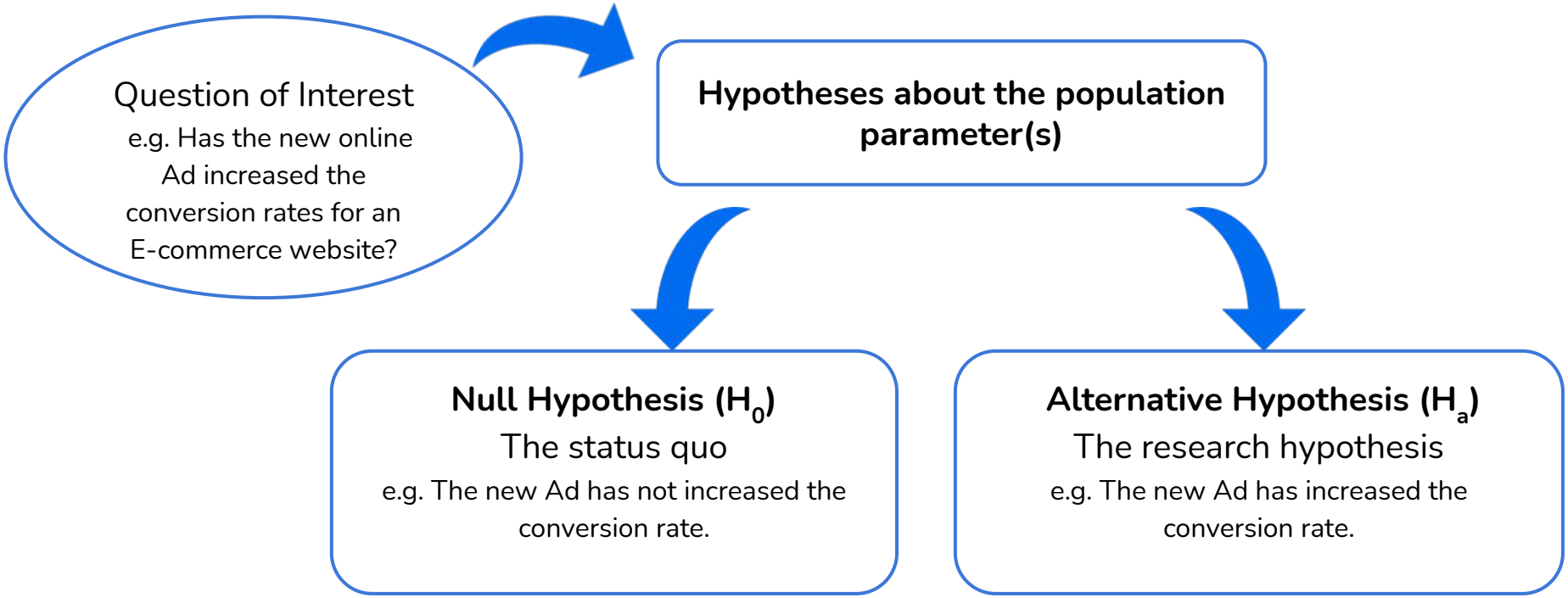
MIT ADSP

Topics

- Hypothesis testing
  - Hypothesis Formulation
  - One Tailed Test vs Two Tailed Test
  - Type I and Type II Errors
- Data Exploration and Networks
  - Testing issues and correction methods
  - Dimensionality Reduction
  - Data analysis using Graph Networks
- Unsupervised Learning
  - Different clustering Algorithms
- Regression
  - Simple Linear and Multiple Regression
  - Overfitting , Bias variance tradeoff and Regularisation
  - Cross validation and Bootstrapping
- Classification
  - Logistic Regression and KNN
  - LDA and QDA
  - Performance Measures Classification

Hypothesis Testing

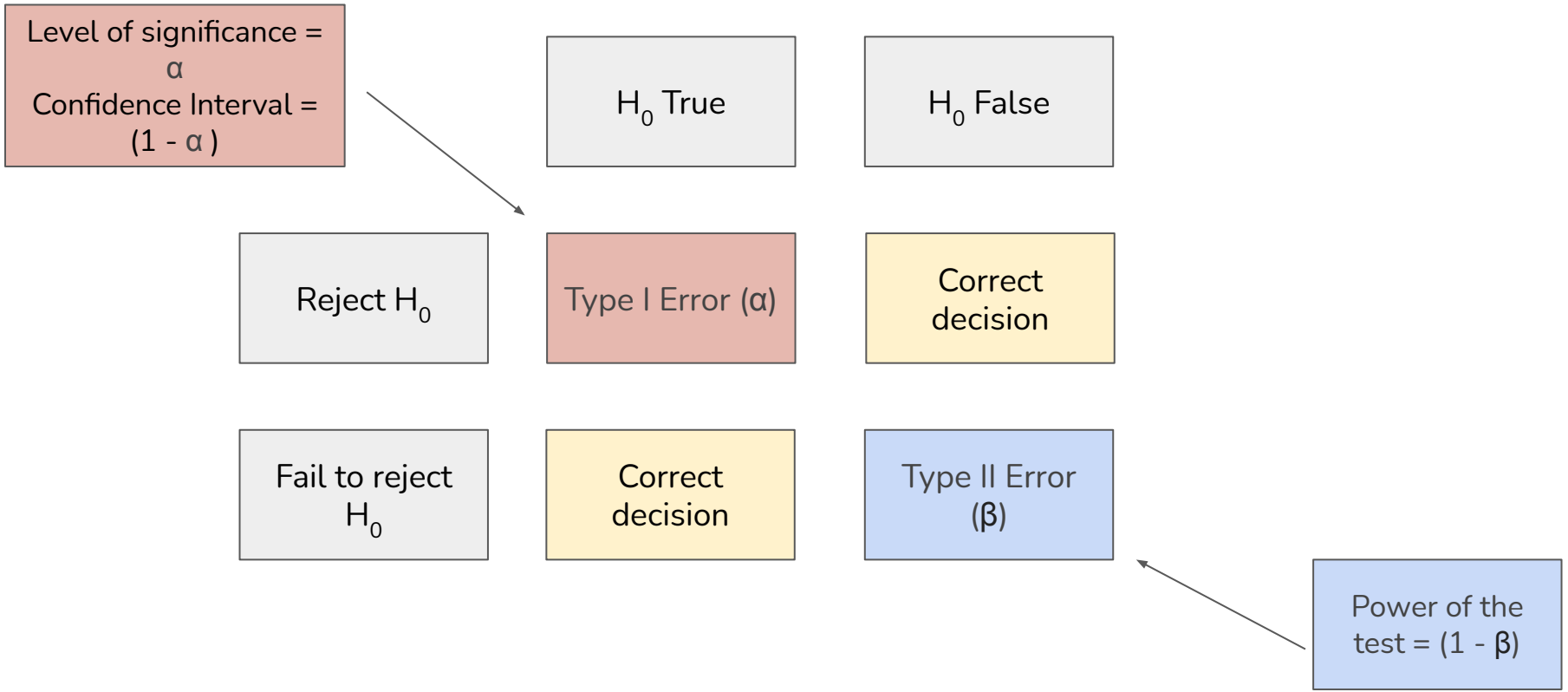
Introduction to Hypothesis Testing



Key terms in Hypothesis Testing

P-Value	• Probability of observing equal or more extreme results than the computed test statistic, under the null hypothesis. • The smaller the p-value, the stronger the evidence against the null hypothesis.
Level of Significance	• The significance level (denoted by α), is the probability of rejecting the null hypothesis when it is true. • It is a measure of the strength of the evidence that must be present in the sample data to reject the null hypothesis.
Acceptance or Rejection Region	• The total area under the distribution curve of the test statistic is partitioned into acceptance and rejection region • Reject the null hypothesis when the test statistic lies in the rejection region, else we fail to reject it
Types of Error	• There are two types of errors - Type I and Type II

Type I and Type II errors



Let's go through an example

Problem Statement: The store manager believes that the average waiting time for the customers at checkouts has become worse than 15 minutes. Formulate the hypothesis.

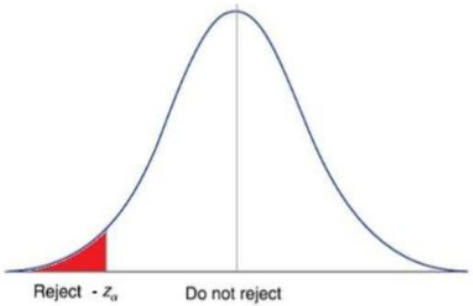
Null Hypothesis (H_0): The average waiting time at checkouts is less than equal to 15 minutes.

Alternate Hypothesis (H_a): The average waiting time at checkouts is more than 15 minutes.

Type I error (false positive): Reject Null hypothesis when it is indeed true. "The fact is that the average waiting time at checkout is less than equal to 15 minutes but the store manager has identified that it is more than 15 minutes".

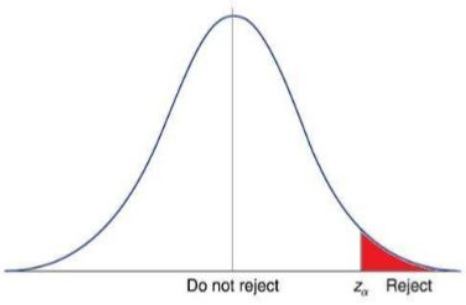
Type II error (false negative): Fail to reject Null hypothesis when it is indeed false. "The fact is that the average waiting time at checkout is more than 15 minutes but the store manager has identified that it is less than equal to 15 minutes".

One-tailed vs Two-tailed Test



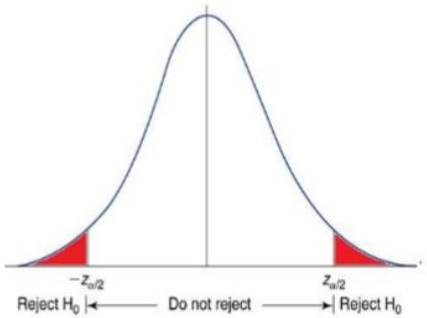
- Lower tail test.
- H_1: $\mu <$ .....

Reject H_0 if the value of test statistic is too small



- Upper tail test.
- H_1: $\mu >$ .....

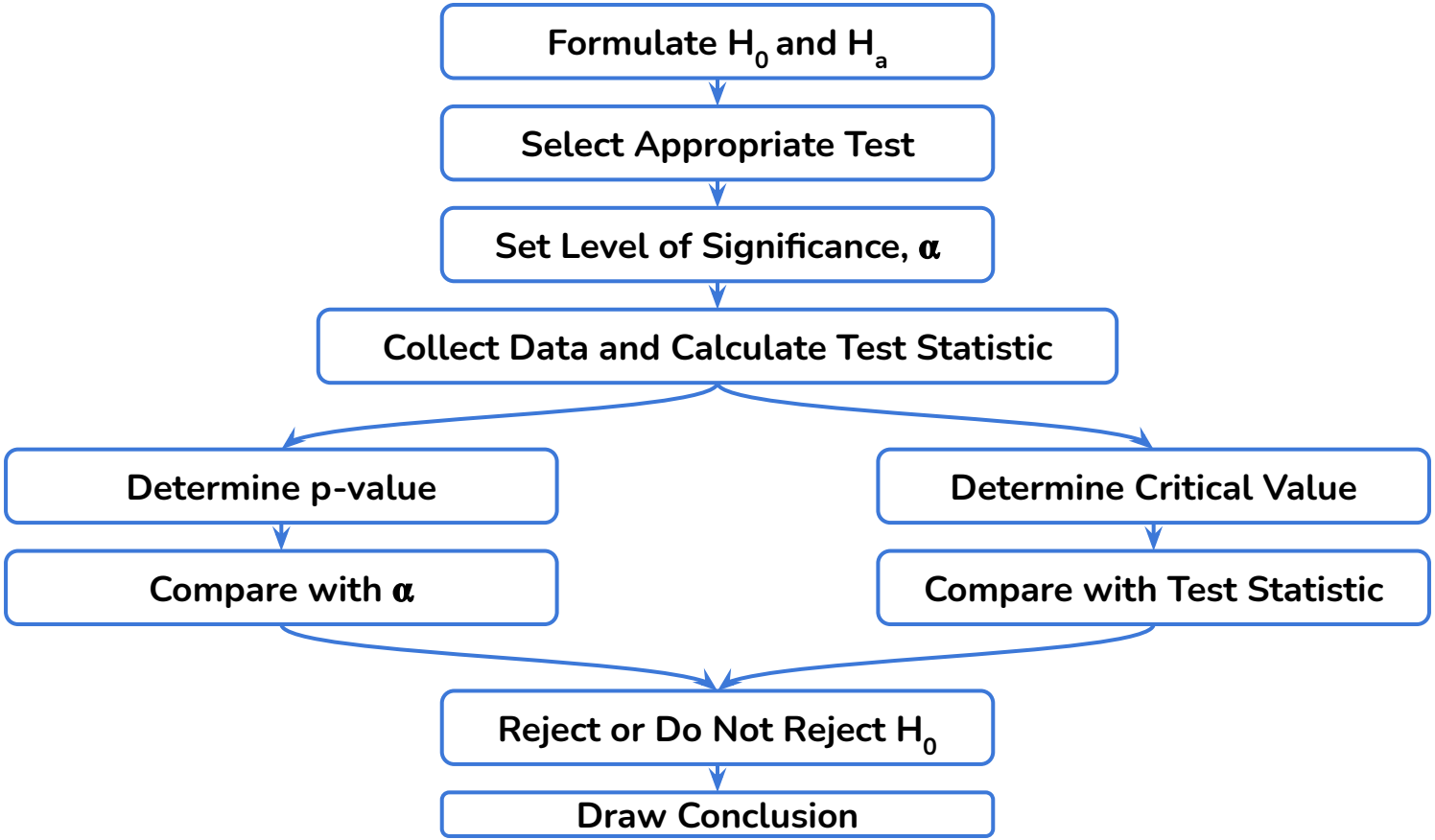
Reject H_0 if the value of test statistic is too large



- Two tail test.
- H_1: $\mu \neq$ .....

Reject H_0 if the value of test statistic is either too small or too large

Hypothesis Testing Steps



Data Exploration and Networks

Back to 1st page

Contents

1. Multiple testing issues and corrections
2. Need for dimensionality reduction
3. PCA-t-SNE and their differences
4. Graph
5. Adjacency Matrix
6. Degree and its calculation
7. Centrality Measures

Multiple testing issue and their corrections

Multiple testing problem

- This problem arises when multiple hypothesis are tested simultaneously.
- The number of false positives increases as you test more number of hypotheses

Following are the correction methods that can be used to deal with this problem:

- **Bonferroni correction**
  - It states that the corrected significance level for all the test combined is **α/m.** where **m** is the total number of hypothesis tests performed
  - Reject null hypothesis H0 when **p-value ≤ α/m or m*p-value ≤ α**
- **Holm-Bonferroni correction**
  - Sort p-values in increasing order: p(1) ≤ · · · ≤ p(m), The corrected significance level for the ith test is **α/(m-i+1).**
  - Reject null hypothesis H0 **p(i) ≤ α/(m-i-1) or (m – i + 1)*p(i) ≤ α**
- **Benjamini-Hochberg correction:**
  - Sort p-values in increasing order: p(1) ≤ · · · ≤ p(m), The corrected significance level for the ith test is **α*i/(m).**
  - Reject null hypothesis H0 **p(i) ≤ α*i/(m) or m*p(i)/i ≤ α**

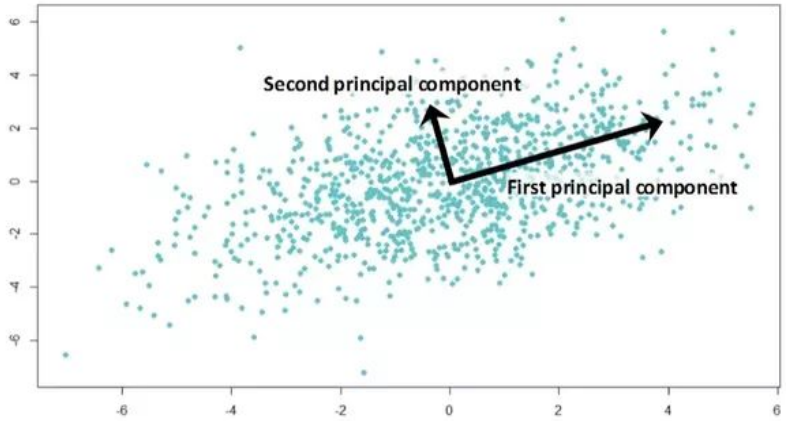
Need for dimensionality reduction

- Dimensionality reduction is the process to reduce the number of dimensions in the feature space.
- In the machine learning, we tend to add many features to get more accurate results. However, after a certain point the performance and robustness of the model starts decreasing and computational complexity starts increasing as we increase the number of features. This is called curse of dimensionality where the sample density decrease exponentially with the increase of dimensionality.
    - We use dimensionality reduction to transform the data into low dimensions while keeping most of the information intact.
    - It also helps us to visualize the high dimensional data to 2D & 3D.
- There are the following techniques we can use for dimensionality reduction:
    - PCA
    - t-SNE

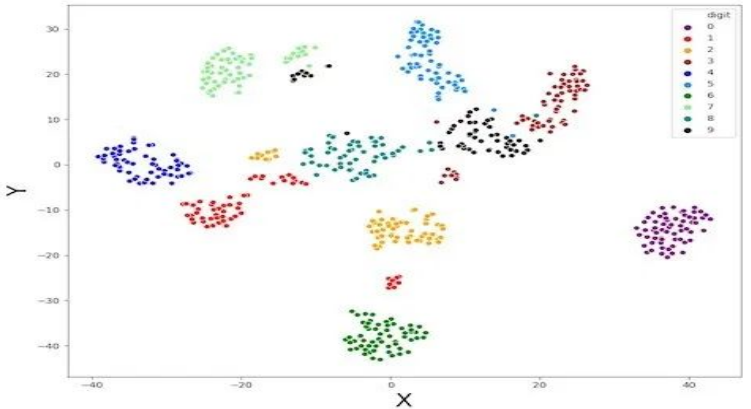
PCA and t-SNE

Principal component analysis (PCA) is a dimensionality reduction technique used for identification of a smaller number of uncorrelated variables known as principal components from a larger dataset. The technique is widely used to emphasize variation and capture strong patterns in a dataset.

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets.



PCA



t-SNE

Image Source

Image Source

Difference between PCA and t-SNE

PCA	t-SNE
It tries to capture linear structure in the data.	It tries to capture non-linear structure in the data.
It focuses to preserve the global structure of the data	It focuses to preserve the local structure (i.e. clusters) of the data
There are no hyperparameters involved in PCA	There are some hyperparameters like perplexity, no. of dimensions etc. in t-SNE
PCA works by separating points as far as possible based on the highest variance	t-SNE works by grouping points as close as possible based on the characteristics of the point
It might easily get affected by outliers	It can handle outliers as well

Why do we study graphs and networks

Graph is basically the study of relationships. It has certain nodes (vertices) and links (edges) that creates these relationships.

It can be used to model create many types of relations and processes in physical, social, biological and information system, and has a wide range of applications:

- Community networks (through social media)
- Google maps
- DNA/RNA sequencing
- Search engine rankings

Example: This friendship network shows us a bunch of friends, the networks they belong to, and the social cliques they are part of.

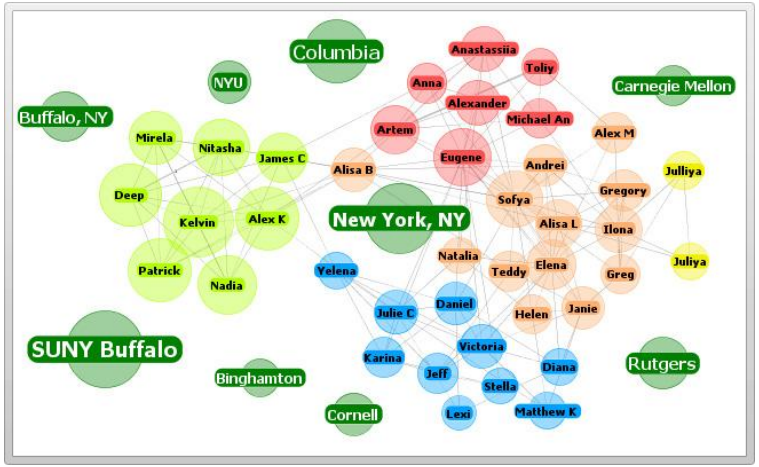


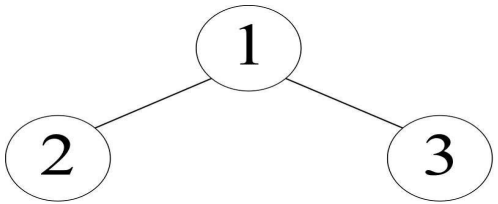
Image Source

Adjacency Matrix

We can represent the graph as an adjacency matrix, where the row and column indices represent the nodes, and the entries in the matrix represent the absence or presence of an edge between the nodes.

Example: For a graph 2-1-3, the adjacency matrix will be-

$$\begin{pmatrix} 0,1,1 \\ 1,0,0 \\ 1,0,0 \end{pmatrix}$$



Adjacency matrix is the best representation of a graph into a mathematical form that tells us whether there are any edges between all sets of nodes. The diagonal of this matrix will always be zero if there are no self loops in the network.

Degree and its calculation

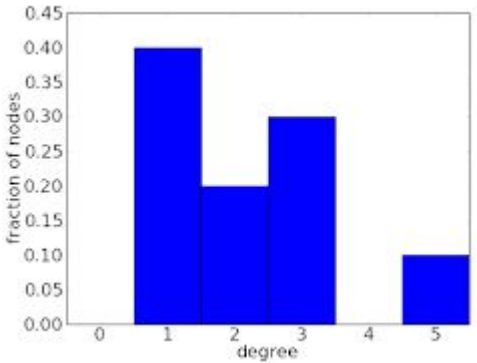
The degree of a node refers to the number of edges that are connected to it. In a directed graph, you can calculate the in-degree and out-degree which means incoming and outgoing connections of a node.

In simple words, it is a popularity measure. The higher the degree, more central the node is.

We can calculate the average degree of a network by using the formula 2m/n, where m is the number of edges and n is the number of nodes.

Degree distribution: It is a probability that the random chosen node has k number of connections.

Here in this graph you can how the degree is varying with the fraction of nodes.



Centrality measures

Centrality measures capture the importance of a node's position in a network. There are the following types of centrality measures:

1. **Degree centrality:** It is a measure of popularity of a node in a network. It does not capture the quality vs quantity.
2. **Propagated degree (eigenvector) centrality:** It measures the importance of a node in a graph with respect to the importance of its neighbors. If a node is connected to highly important nodes, it will have a higher score as compared to a node which is connected to less important nodes.
3. **Closeness centrality:** It tracks how close a node is to another by measuring the distance between them. In other words, it measures the node efficiency in terms of connection to other nodes.
4. **Betweenness centrality:** It measures the importance of a node in a network based upon how many times it occurs in the shortest path between all pairs of nodes in a graph. Basically, It measures the extent to which a node lies on paths between other nodes.

Real life example of a network and its centrality measures

In a **social network**

- High degree centrality - most popular person who can quickly connect with the wider network
- High eigenvector centrality - most popular person who has good social network with other popular person
- High closeness centrality - person who can influence the whole network most quickly
- High betweenness centrality - person who influence the flow around the network i.e. removal of those person can break the network

Back to 1st page

Unsupervised Learning

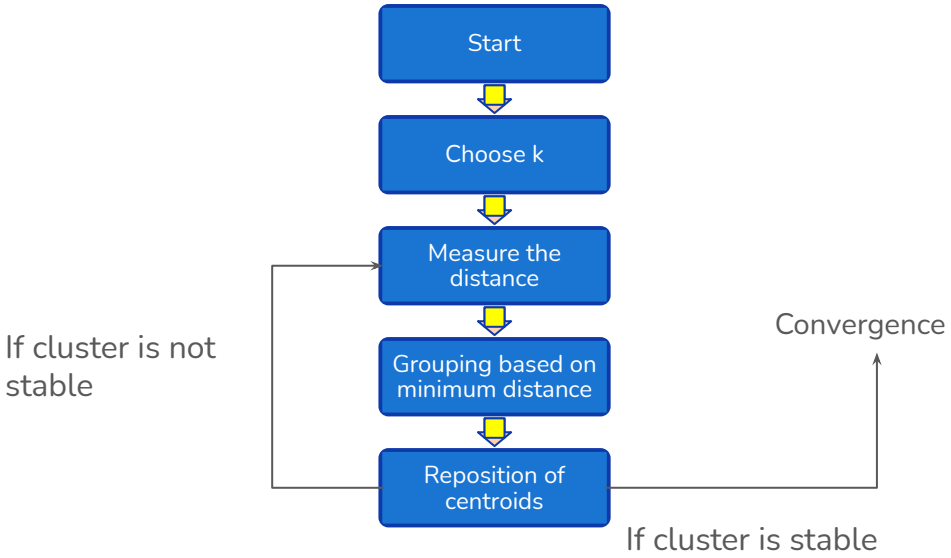
Back to 1st page

Contents

1. K-Means Clustering
2. PAM (K-Medoids) Clustering
3. Hierarchical Clustering
4. GMM
5. DBSCAN

K-Means Clustering

K-Means Clustering is an iterative **algorithm** that divides the unlabeled dataset into **k** different **clusters** in such a way that each point in the dataset belongs to only one group that has similar properties.



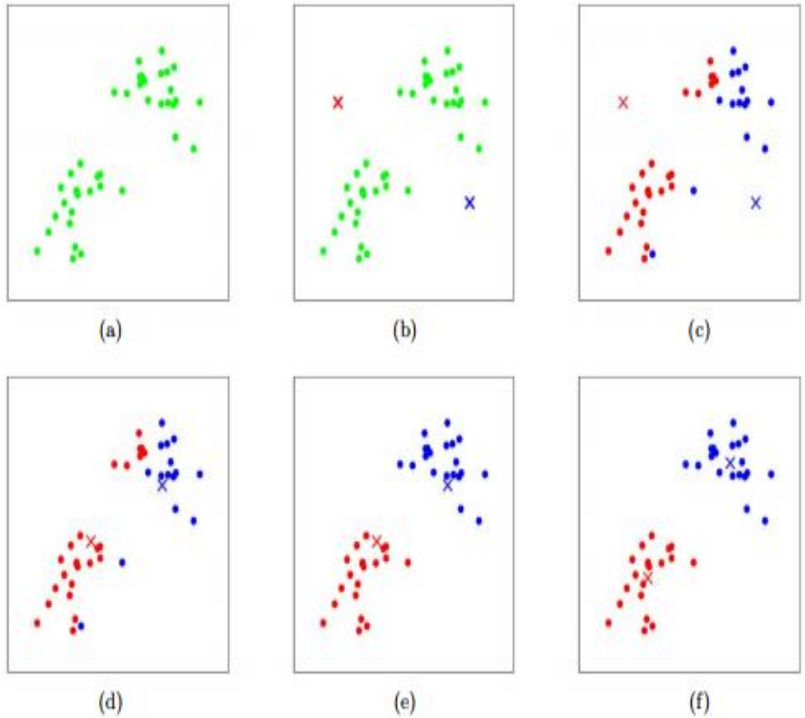


Image Source

Advantages and Disadvantages of using K-Means clustering

Advantages:

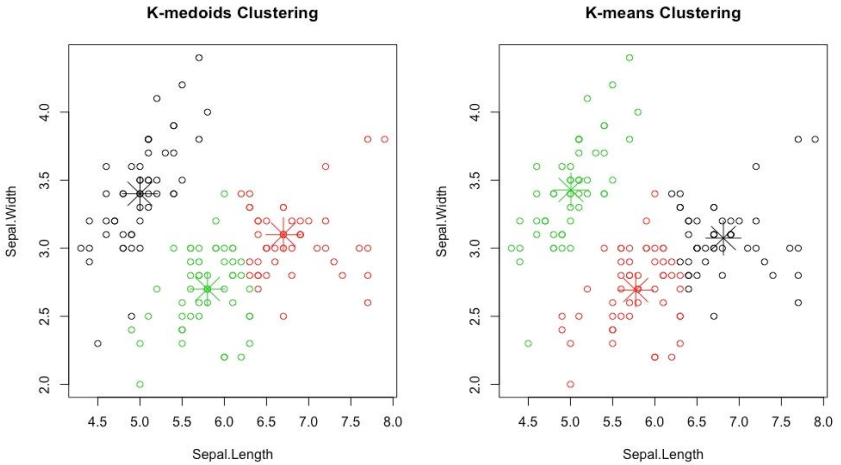
- K-Means is relatively simple to implement
- It scales to large datasets
- It also guarantees convergence
- It can easily adapt to new examples

Disadvantages:

- It is difficult to identify the value of k
- K-Means has trouble clustering data where clusters are of varying sizes and density
- It can easily get affected by outliers
- It assumes data shape to be spherical in nature and does not perform well on the arbitrary data
- It depends on the initial values assigned to the centroids and gives different results for different initialization

Alternative to K-Means - PAM (K-Medoids) Clustering

- The problem with K-Means is that the final centroids are not interpretable i.e. centroids are not actual points but the means of the points present in the cluster.
- The idea behind K-Medoids clustering is to make the final centroids as actual data points so that they are interpretable.
- In K-Medoids, we only change one step from K-Means which is to update the centroids. In this process if there are m points in a cluster, swap the previous centroids with all other (m-1) points from the cluster and finalize the point as new centroid which has minimum loss.
- Because of this, unlike K-Means it is robust to outliers and converges fast.
- You can see in this image that the centroids in K-Medoids are the actual data points represented as the cross, unlike K-Means.



Expectation maximization in GMM Clustering

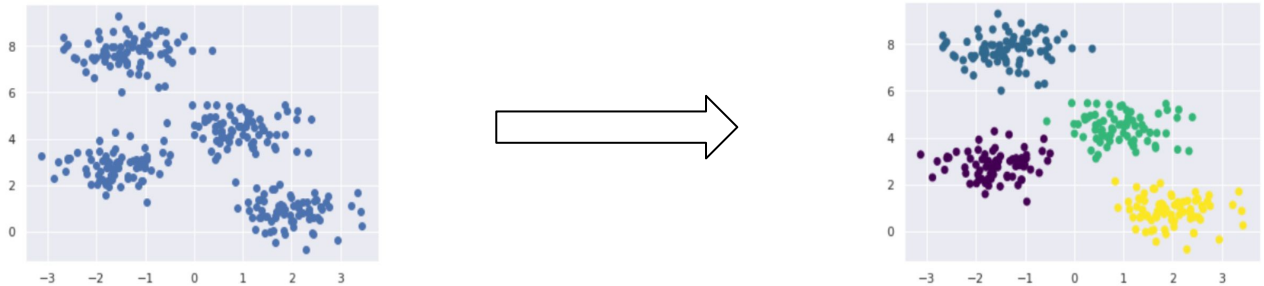
In GMM, we need the parameters of each Gaussian (variance, mean etc.) in order to cluster our data but we need to know which sample belongs to what Gaussian in order to estimate those very same parameters.

That is where we need EM algorithm. There are two steps involved in this algorithm:

1. **The E-step:** It estimates the probability that a given observation to be in a cluster/distribution. This value will be high when the point is assigned to the right cluster and lower otherwise.
2. **The M-step:** In this step we want to maximize the likelihood that each observation came from the distribution

After that we reiterate these two steps and updates the probabilities of an observation to be in a cluster.

Example of GMM clustering



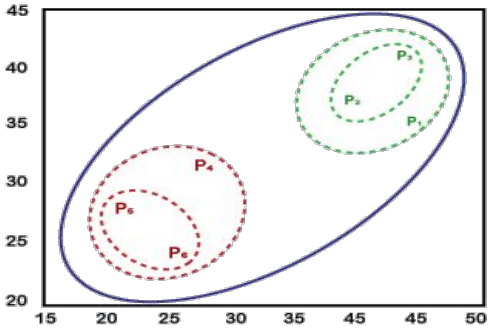
Hierarchical Clustering

Hierarchical clustering is an unsupervised clustering algorithm which involves creating clusters that have predominant ordering from top to bottom. For e.g: All files and folders on our hard disk are organized in a hierarchy.

The algorithm groups similar objects into groups called ***clusters***. The endpoint is a set of clusters or groups, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

Steps

- Make each data point a single-point cluster → forms N clusters
- Take the two closest data points and make them one cluster → forms N-1 clusters
- Take the two closest clusters and make them one cluster → Forms N-2 clusters.
- Repeat step-3 until you are left with only one cluster.



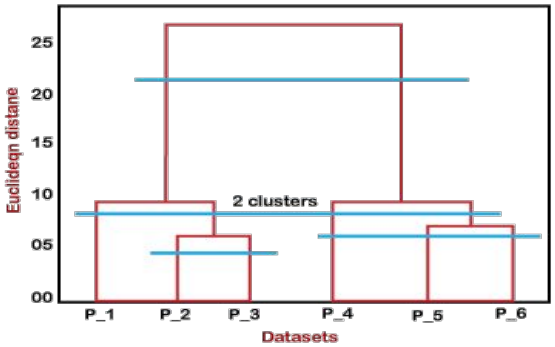


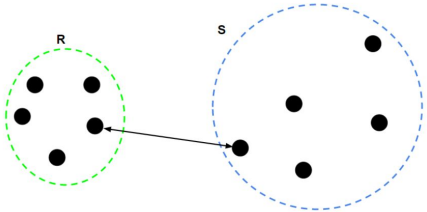
Image source

Dissimilarity among clusters in hierarchical clustering

There are following ways by which we can measure dissimilarity among clusters in hierarchical clustering:

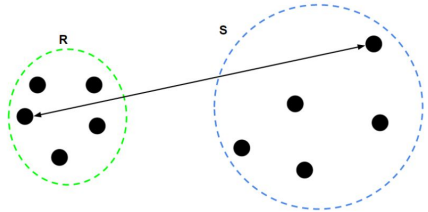
- **Single linkage:** It measure the closest pair of points i.e the minimum distance.

  L(R,S) = min(d(i,j)) where i belongs to R and j belongs to S



- **Complete linkage:** It measure the farthest pair of points i.e the maximum distance.

  L(R,S) = max(d(i,j)) where i belongs to R and j belongs to S



- **Average linkage:** It measure the average dissimilarity over all pairs i.e. the average distance

$$L(R,S) = \frac{1}{n_R+n_S} \sum_{i=1}^{n_R} \sum_{j=1}^{n_S} D(i,j), i \epsilon R, j \epsilon S$$

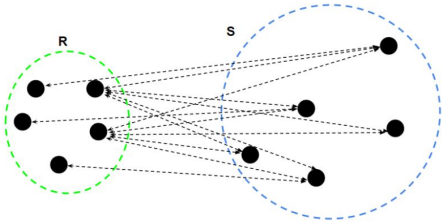


Image Source

DBSCAN

DBSCAN stands for **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise.

It recognizes groups in the data by looking at the local density of a data point. Unlike K-means, **DBSCAN clustering is not sensitive to outliers** and also does not require the number of clusters to be told beforehand.

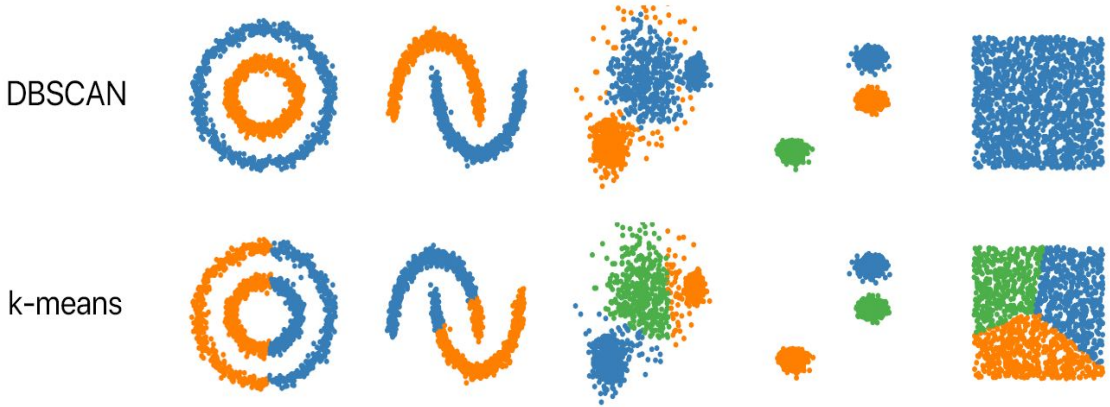
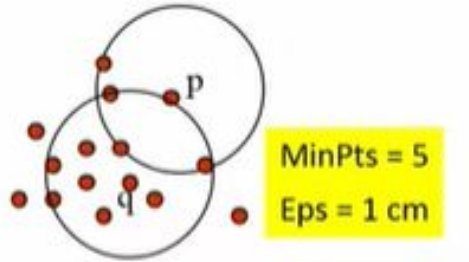


Image Source

Parameters in DBSCAN

- **eps (‘ε’):** It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to ‘eps’ then they are considered as neighbors. If the eps value is chosen too small then a large part of the data will be considered as outliers. If it is chosen very large then the clusters will merge and majority of the data points will be in the same clusters. One way to find the eps value is based on the k-distance graph



- **MinPts:** Minimum number of neighbors (data points) within eps radius. Larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least

Back to 1st page

Regression

Back to 1st page

Contents

Statistics vs Machine Learning

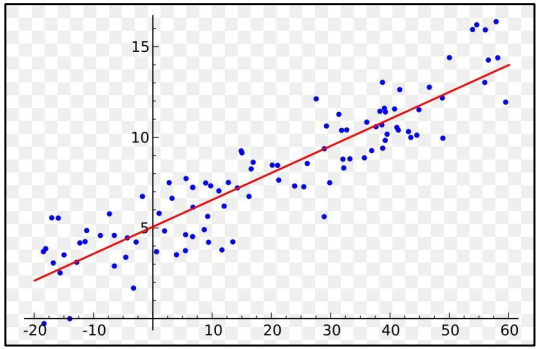
The difference between machine learning and statistics is their purpose. Machine learning models are designed to make the most accurate predictions possible. Statistical models are designed for inference about the relationships between variables.
The following table provides the major differences between statistics and the machine learning point of view:

Statistics	Machine Learning
Emphasis on deep theorems on complex models	Emphasis on the underlying algorithm
Focus on hypothesis testing and interpretability	Focus on predicting accuracy of the model
Inference on parameter estimation, errors and predictions	Inference on prediction
Deep understanding of simple models	Theory does not always explain success

Linear Regression

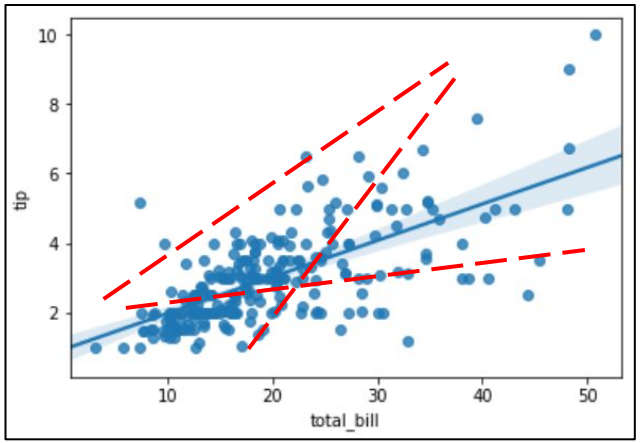
- Linear regression is a way to identify a relationship between the independent variable(s) and the dependent variable
- We can use these relationships to predict values for one variable for given value(s) of other variable(s)
- It assumes the relationship between variables can be modeled through linear equation or an equation of line.
- The variable, which is used in prediction is termed as independent/explanatory/regressor where the predicted variable is termed as dependent/target/response variable.
- In case of linear regression with a single explanatory variable, the linear combination can be expressed as :

response = intercept + constant * explanatory variable



Best fit line in the linear regression model

- Learning from the data, the model generates a line that fits the data.
- Our aim is to find a regression line that best fits the data
- By best fit, it means that the line will be such that the cumulative distance of all the points from the line is minimized
- Mathematically, the line that minimizes the sum of squared error of residuals is called Regression Line or the Best Fit Line.



In the example here, you can see a scatter plot between the *tip* amount and the *total_bill* amount

We can see that there is positive correlation between these two - as the bill amount increases, the tip increases

The line in blue that you see is the 'best fit' line - those in red are some examples of all other lines that are not the 'best fit'

What is Multiple Linear Regression?

- This is just the extension of the concept of simple linear regression with one variable
- In the real world, any phenomenon or outcome could be driven by many different independent variables
- Therefore there is a need to have a mathematical model that can capture this relationship

  - **Ex:** Predicting the price of a house, we need to consider various attributes such as area, number of rooms, number of kitchens etc. Such a regression problem is an example of multiple linear regression.
  - The equation for multiple linear regression can be represented by :

target = intercept + constant 1*feature 1 + constant 2*feature 2 + constant 3*feature 3 + ...

- The model aims to find the constants and intercept such that this line is the best fit

Regression Model Evaluation Metrics

R-squared	Adjusted R-squared	Mean Absolute Error	Root Mean Square Error
• Measure of the % of variance in the target variable explained by the model • Generally the first metric to look at for linear regression model performance • Higher the better	• Conceptually, very similar to R-squared but penalizes for addition of too many variables • Generally used when you have too many variables as adding more variables always increases R^2 but not Adjusted R^2 • Higher the better	• Simplest metric to check prediction accuracy • Same unit as dependent variable • Not sensitive to outliers i.e. errors doesn't increase too much if there are outliers • Difficult to optimize from mathematical point of view (pure maths logic) • Lower the better	• Another metric to measure the accuracy of prediction • Same unit as dependent variable • Sensitive to outliers - errors will be magnified due to square function • But has other mathematical advantages that will be covered later • Lower the better

$$R^2 = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - yhat_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(N-1)}{N-p-1}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - yhat_i|$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - yhat_i)^2}$$

Assumptions of Linear Regression

Assumption	How to test	How to fix
There should be a linear relationship between dependent and independent variables	Pairplot / Correlation of each independent variables with dependent variable	Transform variables that appear non-linear (log, square root etc.)
No multicollinearity in independent variables	Heatmaps of correlations or VIF (Variance inflation factor)	Remove correlated variables or merge them
No Heteroskedasticity - residuals should have constant variance	Plot residuals vs. fitted values and check the plot	Non-linear transformation of dependent variable or add other important variables
Residuals must be normally distributed	Plot residuals or use Q-Q plot	Non-linear transformation of independent or dependent variable

Bias-Variance: Underfitting and Overfitting

Bias: Bias is the difference between the prediction of our model and the correct value which which we are trying to predict. Model with high bias gives less attention to the training data and overgeneralize the model which leads to high error on training and test data.

Variance: Variance is the value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the test data. Therefore, such models perform very well on training data but has high error on test data

In supervised learning, **underfitting** happens when a model is not able to capture the underlying pattern of the data. These models usually have high bias and low variance whereas, **overfitting** happens when our model captures the noise along with the underlying pattern in data. These models usually have low bias and high variance.

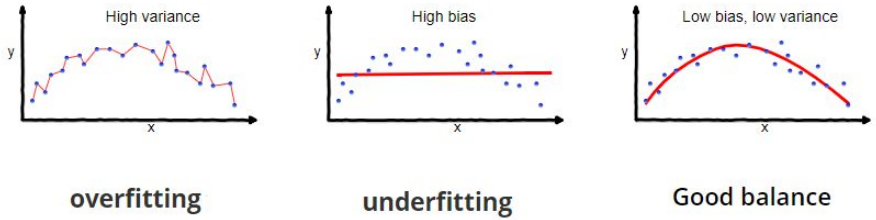


Image Source

Bias-Variance Tradeoff

If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if our model has large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.

An optimal balance of bias and variance would never overfit or underfit the model.

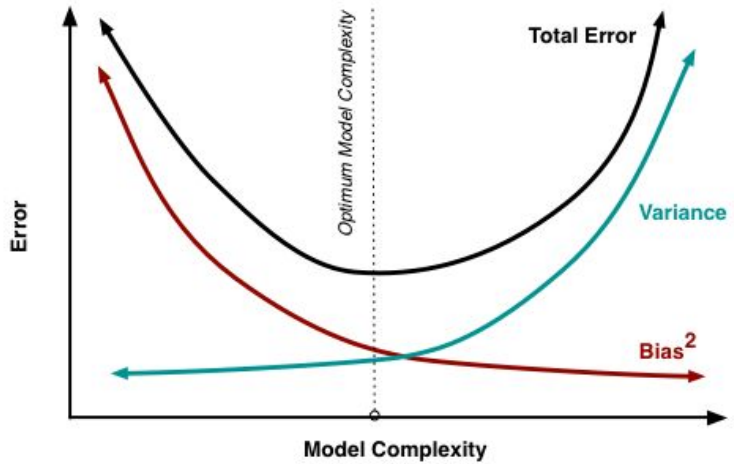


Image Source

Regularization and its types

- Regularization is the process which regularizes or shrinks the coefficients towards zero. In other words, this technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.
- Regularization, significantly reduces the variance of the model, without substantial increase in its bias.
- There are two types of regularization:
    - **Lasso Regression:** In this technique we add $\alpha * \sum|\beta|$ as the shrinkage quantity. It only penalizes the high coefficients. It has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large. This technique is also called L1 regularization.
    - **Ridge Regression:** In this technique, we modify the residual sum of squares by adding the shrinkage quantity $\alpha * \sum\beta^2$ and use α as the tuning parameter that decides how much we want to penalize the flexibility of our model. This technique is also called L2 regularization.

Cross-validation and its types

Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

- It provides some kind of assurance that your model has got most of the pattern from the data set correct and it is not picking up some noise.
- We will be discussing two types of cross validation techniques -
    2. K-Fold Cross-validation
    3. Leave-One-Out Cross-validation (LOOCV)

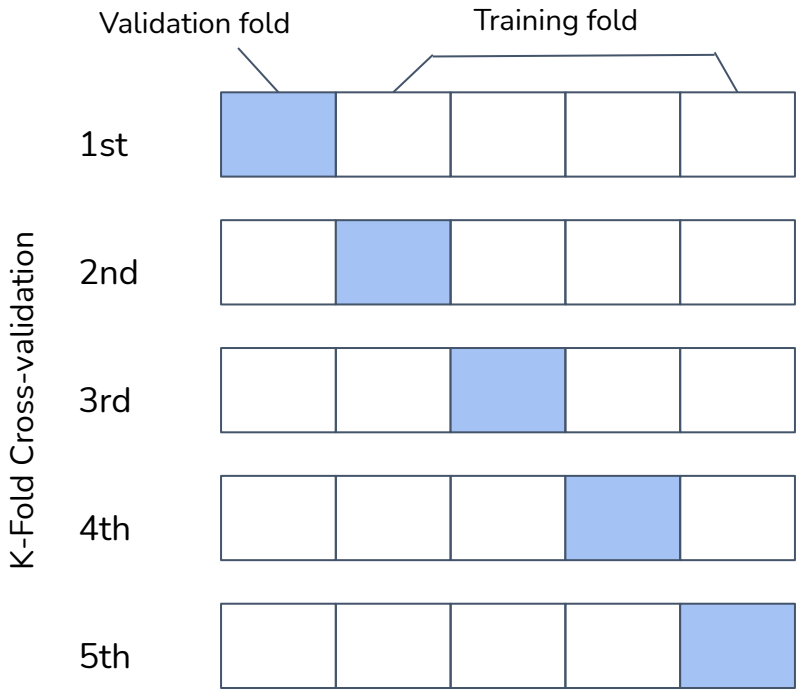
K-fold Cross-validation

This algorithm has a single parameter called K that refers to the number of groups that a given data sample is to be split into.

This algorithm has following procedure:

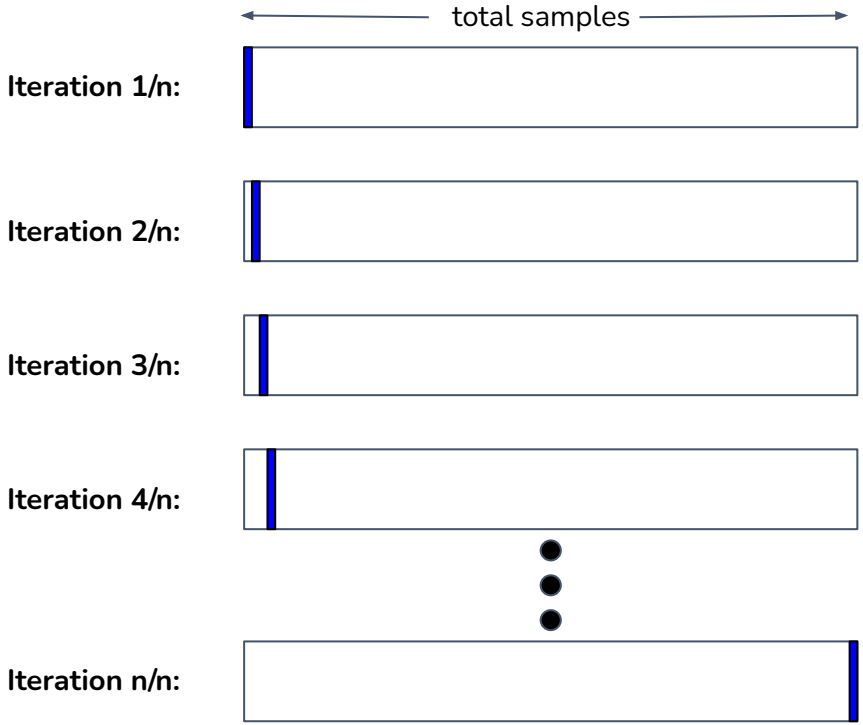
1. Shuffle the dataset randomly.
2. Split the whole dataset into K groups
3. For each unique group, take one as a hold out set and remaining as training set.
4. Repeat the step 3, for all groups
5. Summarize the skill of the model using the sample of model evaluation scores of all groups

Here, K = 5



Leave-One-Out Cross-validation (LOOCV)

- LOOCV is a special case of K-fold cross validation where K equals n, n being the number of data points in the sample.
- This approach leaves 1 data point out of the training data, i.e. if there are n data points in the original sample then, n-1 samples are used to train the model and p points are used as the validation set.
- This is repeated for all combinations in which the original sample can be separated this way, and then the error is averaged for all trials, to give overall effectiveness.
- The number of possible combinations is equal to the number of data points in the original sample or n.



Bootstrapping

Bootstrapping (also called Bootstrap sampling) is a resampling method that involves drawing of sample data repeatedly with replacement to estimate a population parameter.

It involves the following steps:

1. Choose a number of bootstrap samples to perform
2. Choose a sample size n
3. For each bootstrap sample
    1. Draw a sample with replacement with the chosen size
    2. Calculate the statistic on the sample
4. Calculate the mean of the calculated sample statistics.

Bootstrap sampling can be used to estimate the parameter of a population (i.e. mean, standard error etc.)

Back to 1st page

Classification

Back to 1st page

Contents

- LDA vs QDA
- Logistic Regression
- Confusion Matrix
- Why accuracy is not a good performance measure
- Thresholding using Precision-Recall Curve
- F1-score
- KNN

LDA vs QDA

Linear Discriminant Analysis	Quadratic Discriminant Analysis
It is a linear classifier but much less flexible than QDA	It is a non-linear classifier but more flexible than LDA
It assumes a common covariance matrix for all the classes	It assumes that each class has its own covariance matrix
It is prefered when the training set has a few observations	It is prefered when the training set is very large.
It can be used as a dimensionality reduction technique	It can not be used as a dimensionality reduction technique

Why do we use Logistic Regression?

- Logistic Regression is a supervised learning algorithm which is used for the classification problems i.e. where the dependent variable is categorical
- In logistic regression, we use the sigmoid function to calculate the probability of the dependent variable
- The real life applications of logistic regression are churn prediction, spam detection etc.
- The below image shows how logistic regression is different from linear regression in fitting the model

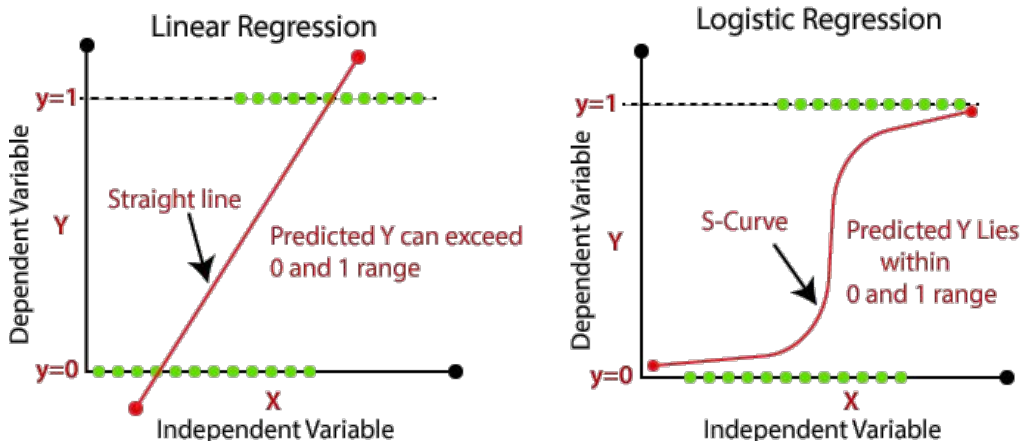


Image Source

Confusion Matrix

It is used to measure the performance of a classification algorithm. It calculates the following metrics:

1. **Accuracy:** Proportion of correctly predicted results among the total number of observations

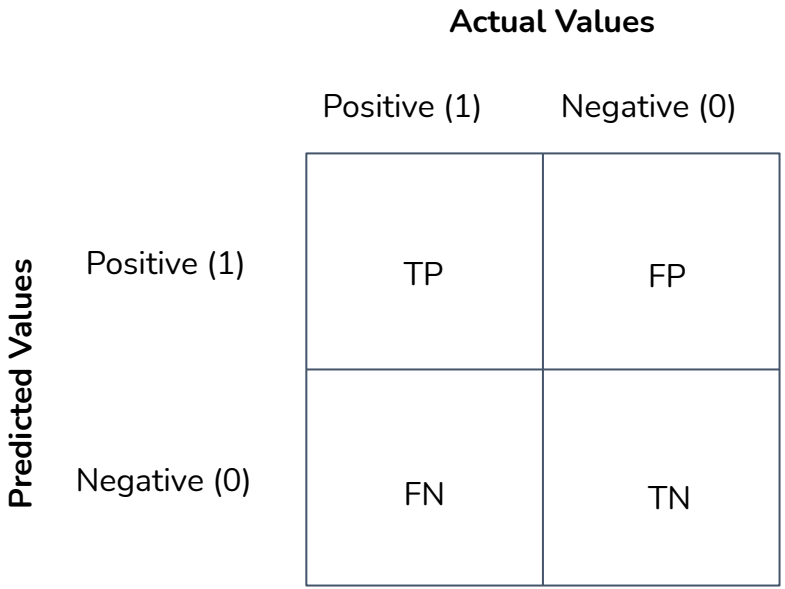
   Accuracy = (TP+TN)/(TP+FP+FN+TN)

1. **Precision:** Proportion of true positives to all the predicted positives i.e. how valid the predictions are

   Precision = (TP)/(TP+FP)

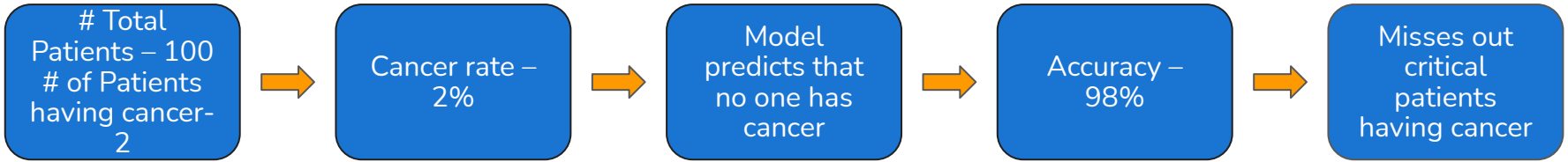
1. **Recall:** Proportion of true positives to all the actual positives i.e. how complete the predictions are

   Recall = (TP)/(TP+FN)



Why accuracy is not always a good performance measure

Accuracy is simply the overall % of correct predictions and can be high even for very useless models.

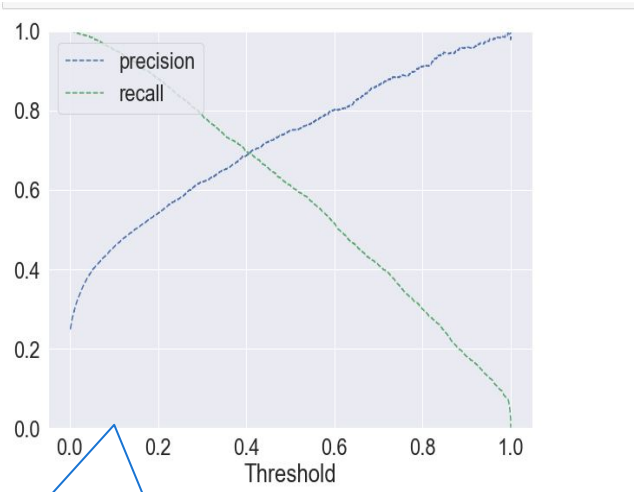


- Here Accuracy will be 98%, even if we predict all patients do not have cancer.
- In this case, Recall should be used as a measure of the model performance; high recall implies fewer false negatives
- Fewer false negatives implies a lower chance of 'missing' a cancer patient i.e. predicting a cancer patient as one not having cancer.
- This is where we need other metrics to evaluate model performance.

- The other metrics are Recall and Precision
  - Recall - What % of actuals 1s did I capture in my prediction?
  - Precision - What % of my predicted 1s are actual 1s?
- There is a tradeoff - as you try to increase Recall, Precision will reduce and vice versa
- This tradeoff can be used to figure out the right threshold to use for the model

How to chose thresholds using the Precision-Recall curve?

- Precision-Recall is a useful measure of success of prediction when the classes are imbalanced.
- The precision-recall curve shows the tradeoff between precision and recall for different thresholds.
- It can be used to select an optimal threshold as required to improve the model performance
- Here as we can see, precision and recall are the same when the threshold is 0.4
- If we want higher precision, we can increase the threshold
- If we want higher recall, we can decrease the threshold



Choosing a threshold can completely change the model performance assessment
It is important to think about what constitutes the 'sweet spot'

Is there a performance measure that can cover both Precision and Recall?

- F1 Score is a measure that takes into account both Precision and Recall.
- F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

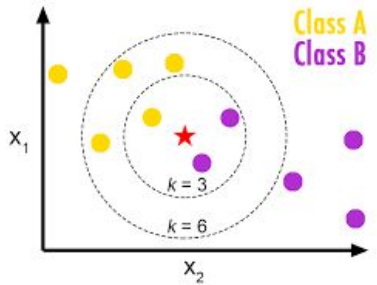
- The highest possible value of F1 score is 1, indicating perfect precision and recall, and the lowest possible value is 0.

K-Nearest Neighbours (kNN) algorithm

This algorithm uses similar features to predict the values of new data points, which means the new data point will be assigned a value based on how similar it is to the data points in the training set. We can define its working in the following steps –

- Step 1 – We need to choose the value of k i.e. the nearest data points. k can be any odd positive integer.
- Step 2 – For each point in the test data do the following –
  - Calculate the distance between the test point and each training point with the help of any of the distance methods, namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate this distance is the Euclidean method.
  - Now, based on the distance value, sort them in ascending order.
  - Next, it will choose the top k rows from the sorted array.
  - Now, it will assign a class to the test point based on the most frequent class of these rows.
- Step 3 – Repeat this process until all the test points are classified in a particular class

We try different values of k and plot it against the test error. The lower the value of the test error, the better the value of k.



Great Learning

Happy Learning !