Is the data structured enough?
Is the data clean enough to analyze?

Sample Collection || Practical Motivation

Data Preparation || Problem Formulation

Numeric prediction == Regression
Class prediction == Classification
Structure detection == Clustering
Anomaly detection == Anomaly Detection

How to effectively mine the data?
How to compute the vital Statistics?

Exploratory Data Analysis || Statistical Description

How do you summarize the data?
Which vital statistics are relevant?

Analytical Visualization || Pattern Recognition

Is there a mutual dependence?
What is the mutual

How to reduce error in learnings?
How to generalize the algorithms?

Algorithmic Optimization || Machine Learning

Information Presentation || Statistical Inference

Can you estimate the confidence?

Ethical Consideration || Intelligent Decision

Can you optimize the outcomes?

Structured Data: Highly Organized, Easy to Analyze, Clearly defined variable	Numeric Data	Numeric Continuous Variables	Spreadsheets (Excel, CSV) o Standard SQL Databases o Sensors and Devices
	Categorical Data	Factor/Level/Class Variables	Same^ + Binary data is categorical
	Mixed Data	Numeric and Categorical	Same^
	Time Series Data	Numeric with Timestamps	Stock and Equity Markets o Weather Data over Time o Prices and Promotions
	Network Data	Nodes and Connections	Social Networks and Web o Transport Networks (MRT) o Financial Transactions
Unstructured Data: Highly Unorganized, Non-Obvious Variables Context-Sensitive	Text Data	Words, Phrases, Emoticons	Social Networks and Web o Text Messages / WhatsApp o Books, Wikis, Documents
	Image Data	Pixels and Objects	Social Networks and Web o Mobile Phone Cameras o Blogs, Wikis, Documents
	Video Data	Images, Frames, Objects	YouTube and Social Media o Video Messages and Calls o Mobile Phone Cameras
	Voice Data	Voice Signals and Waves	Songs and Social Media o Microphones and Cameras o Recordings

Basic Statistics

Univariate Statistics

Motivation- To find structure in the data

$mean = \bar{x} = \frac{x_1+\cdots+x_n}{n}$

$S.D. = \sqrt{\frac{(x_1-\bar{x})^2+\cdots+(x_2-\bar{x})^2}{n}}$

$Variance = S.D.^2$

$P(x \le x_{Q1}) = 0.25$

$P(x \ge x_{Q3}) = 0.25$

$P(x_{Q1} \le x \le x_{Q3}) = 0.5$

$median = (x_{Q2})$

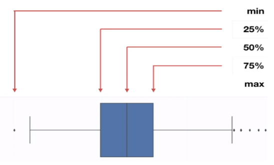
Standard Deviation = average deviation from the mean; measure of dispersion of data

Median = arrange the data in ascending order

Quartiles = Markers to divide the data (25:50:25)

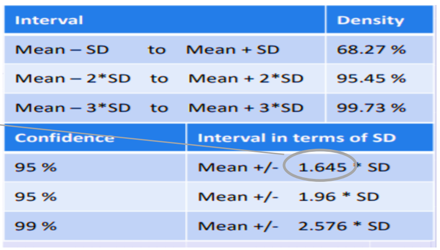
Variance of the data is in unit squared.

Standard deviation increases when the data is more spread out, in comparison to mean absolute deviation (+4, +4, -4, -4 // +7, +1, -6, -2)
Standard deviation enhances the outliers; better to use mean absolute deviation.



For median and quartile, arrange data in ascending order

Whiskers: 1.5*Quartile Gap = 1.5*$(Q_3 - Q_1)$
Density plot/KDE: changes discreet histogram to continuous line
The Normal Distribution= Gaussian Distribution



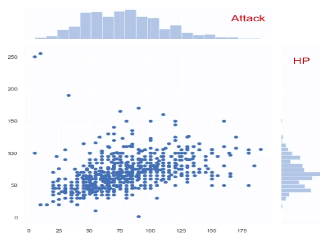
Interval			Density
Mean – SD	to	Mean + SD	68.27 %
Mean – 2*SD	to	Mean + 2*SD	95.45 %
Mean – 3*SD	to	Mean + 3*SD	99.73 %

Confidence	Interval in terms of SD
95 %	Mean +/- 1.645 * SD
95 %	Mean +/- 1.96 * SD
99 %	Mean +/- 2.576 * SD

Called the z values

Bivariate Statistics

Joint Plot



On an average, y increases with x

Correlation Coefficient = R

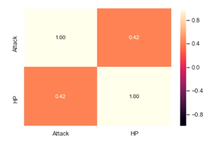
$P_{xy} = \frac{\Sigma(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\Sigma(x_i-\bar{x})^2}\sqrt{\Sigma(y_i-\bar{y})^2}}$

Numerator=covariance
Denominator=St. Dev Product
All points lie exactly on line = correlation 1
Pearson Correlation

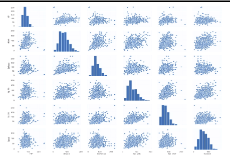
No Dependence	Corr=0
Perfect Positive	Corr=1
Perfect Negative	Corr=-1

Spearman Correlation==Monotonic relationship between two ordinal or cont variables

Correlation Plot / Heat Map



Pair-Plot of Multi Variant Data



Linear Regression

-Objective=Optimal Modelling
- Finding a function of the other variables to predict y
- Supervised Learning in Machine Learning -> splits the data in train and test

Univariate

- **Correlation tells us nothing about causality**

Algorithm:
Guess the initial values of the "parameters" for the hypothesized linear Model.
Predict the values of the Response Variable for all observations in Train data.
Compute the Errors in Train data, compared to actual values of the Response.
Choose a specific Cost Function (like Sum Square of Errors) for Optimization.
Reassign or tune the "parameters" of the model to optimize the cost function.

Cost function we want to minimize:
Residual Sum of Squares (for training)
$J(a,b) = \Sigma(y - a*x - b)^2$ <- ^2 so the + & - errors don't cancel each other
$J(a,b) = \Sigma(y - y_{pred})^2$

Testing the Goodness Fit

Mean Squared Error	Explained Variance (R^2)
$MSE = \frac{1}{n}\Sigma(y - a*x - b)^2$	$R^2 = 1 - \frac{\Sigma(y-a*x-b)^2}{\Sigma(y-y_{mean})^2}$
$MSE = \frac{1}{n}\Sigma(y - y_{pred})^2$	$R^2 = 1 - \frac{\Sigma(y-y_{pred})^2}{\Sigma(y-y_{mean})^2}$
$MSE = \frac{1}{n} * RSS$	$R^2 = 1 - \frac{RSS}{TSS}$ = (correlation)²
$RMSE = \sqrt{MSE}$ (beings the unit back to actual)	$R^2 = 1 - \frac{MSE}{Variance}$
The lower the MSE the better	The higher the R^2 the better
MSE does not lie between 0 and 1	$0 \le R^2 \le TSS$

$Variance = \frac{1}{n} * TSS$

Multivariate

- Plot the correlation of y against all others to see the statistical dependence in a correlation map/ heat map
- Relationship pattern is seen in multivariate joint plot
- The linear line is on a hyperplane if the number of variables is > 2. If =2, then 3D.

$J(a,b) = \Sigma(y - y_{pred})^2$

Never extrapolate linear regression: The confidence interval becomes very large
A 95% confidence interval means that 95% of the data are within the confidence zone

Minimalizing cost function: Gradient descent is an efficient optimization algorithm that attempts to find a local or global minima of a function.
Gradient descent enables a model to learn the gradient or direction that the model should take in order to reduce errors = the RSS.
As the model iterates, it gradually converges towards a minimum where further tweaks to the parameters produce little or zero changes in the loss= convergence. Gradient descent, therefore, enables the learning process to make corrective updates to the learned estimates that move the model toward an optimal combination of parameters.
With a high learning rate we can cover more ground each step, but we risk overshooting the lowest point since the slope of the hill is constantly changing. A low learning rate is more precise, but calculating the gradient is time-consuming, so it will take us a very long time to get to the bottom.

Classification

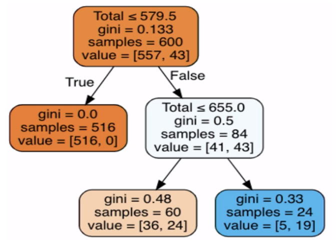
Binary Classification

· Plot the continuous variable as x axis and the categorical data as y-axis in a box plot//swarm-plot

Decision Tree- Uni/Multi Variate

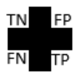
-Objective: Optimal Partitions
- Predicting a response from predictors
Decision of Partition depends on the Gini Index (metric of misclassification)

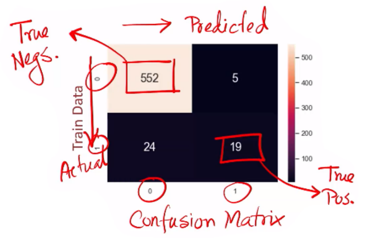
$$gini = \frac{x}{n}\left(1-\frac{x}{n}\right)+\frac{y}{n}\left(1-\frac{y}{n}\right)$$



Testing the prediction
Plot the confusion matrix for train data and test data

$$False\ Positive\ Rate = \frac{FP}{FP+TN}$$
$$False\ Negative\ Rate = \frac{FN}{FN+TP}$$
$$Classification\ Accuracy = \frac{TP+TN}{Total\ Datapoints}$$



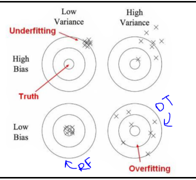


F1 Score: high F1 score = low false positives and low false negatives, so you're correctly identifying real threats and you are not disturbed by false alarms

Multi-Class Classification

Random Forest - Multiple Decision Trees

·Plot multiclass boxplots, one for each predictor
·Partitions made in multiclass data space represented using consecutive binary decision
·Improvement on a single tree: random forest = cancellation of errors = every tree has a different training set + different variables = sufficient randomization

$gini = \sum \frac{x_i}{n}\left(1-\frac{x_i}{n}\right)$			How to improve? -Increase the depth of tree -Merge some of the types -Feature Selection -GPU=parallel processing of trees
Confusion Matrix -The diagonals are correctly classified, rest are wrong			
Choose Variables	All	All	Random
Choose Data Points	All	Random	Random
Create tree (+repeat)	-		-
Tree=	Identical	Similar	Diverse
Collaboration	No	Some	Great

Gradient Boosting

Better than Random Forest because it's a method of converting weak learners into strong learners. Here, a new tree is a fit on a modified version of the original data set. It allows one to optimise a user specified cost function (=more control to the situation).

Recommendation Systems

Tracking Customer Ratings
Customers who bought this item also bought: Tracking Purchase Pattern
Customers who viewed this item also viewed: Tracking Browsing Pattern
Sponsored product related to this item: Learning "Similar" Products
Trending Now/ Popular on Netflix: Promoting popular products
Because you watched 'xyz': Promoting "Similar" Products
Top picks for 'XYZ': Promoting choices of "Similar" Customers
Sparse Data: Baseline Prediction Method:Average of all given ratings
Based on Item-Similarity and Based on User- Similarity
User-Item Matrix
Each of the purchase patterns are taken as vectors, and the distance is calculated

Euclidean Similarity: numeric entries like ratings Small distance means high similarity	$d(x,y) = \sqrt{(x_1-y_1)^2+\cdots+(x_n-y_n)^2}$
Cosine Similarity: same items but different quantities; length of vector will be normalized Small angle (=higher value of the fraction) means high similarity Cos gives us a better sense of how much common there is	$\cos(x,y) = \frac{x\bar{y}}{\lVert\bar{x}\rVert\lVert\bar{y}\rVert}$ $\cos(x,y) = \frac{x_1y_1+\cdots+x_ny_n}{\sqrt{x_1^2+\cdots+x_n^2}\sqrt{y_1^2+\cdots+y_n^2}}$
Jaccard Similarity: entries are binary; who watches what Considers users and their purchase pattern as sets Large Intersection means High Similarity	$J(x,y) = \frac{Set(x)\cap Set(y)}{Set(x)\cup Set(y)}$

Clustering

(Unsupervised Learning)
The notion of distance is important in clustering in order to find the near and far points

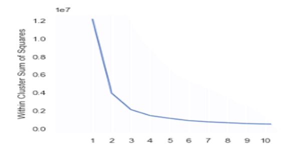
K-Means Clustering

Choose K- The potential number of clusters	Parameter
Choose K cluster centroids from the dataset	Initialization
For each point, re-label according to the nearest centroid	Iteration
For each cluster of data points, re-compute the centroid of the cluster, using the mean	

Problems with K-Means:
- The choice for initial centroids or cluster centers dictate the algorithm.
-----Solution: Use K-Means++: to select 2 cluster centers, select the first one randomly, then points which are farthest to it gets assigned as the second.
- The number of clusters k has to be chosen first, or guessed
-----Solution: run a bunch of k-means algorithms with random starting centroids and take some common clustering result as the final result
- The k-Means algorithm prefers clusters of similar size and shape, i.e. spherical clusters, generally of equal diameter
Optimization Questions → we optimize to avoid finding patterns in noise+compare clus.
1) What is a nice Clustering metric-WSS//BSS?

Within Sum of Squares: Cluster Cohesion → Goal= minimize WSS
In general, a cluster that has a small sum of squares is more compact
As the number of observations increases, the sum of squares becomes larger. Therefore, the within-cluster sum of squares is often not directly comparable across clusters with different numbers of observations. To compare the within-cluster variability of different clusters, use the average distance from centroid instead.

Between Sum of Squares: Cluster Separation → Goal= maximize BSS
SStotal=SSwithin+SSbetween. So, if SSwithin is minimized then SSbetween is maximized.



X-axis max==total number of data points==each point is its own custer== WSS is 0; BSS is max
WSS=n*(Var(x))+n*Var(y)
WSS of min number of clusters=max
Elbow Plot/ Angle Plot

Average Distance form centroid
Clusters that have higher values exhibit greater variability of the observations within the cluster.

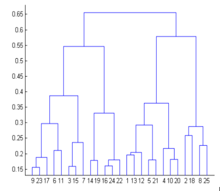
DBSCAN Clustering: DBSCAN defines clusters based on dense areas. There is no point using DBSCAN on 1D data.

Gaussian Mixture Clustering

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters; Unlike K-Means, clusters can be shaped other than spherical clusters.

Hierarchical Clustering

Hierarchical clustering can't handle big data well but K Means clustering can. This is because the time complexity of K Means is linear i.e. O(n) while that of hierarchical clustering is quadratic i.e. O(n2).
K Means is found to work well when the shape of the clusters is hyper spherical (like circle in 2D, sphere in 3D).
Results are reproducible in Hierarchical clustering.
K Means clustering requires prior knowledge of K. But, in hierarchical clustering you can stop at whatever number of clusters you find appropriate.


Dendrogram

Anomaly Detection

1) DBSCAN
2) Local Outlier Factor

Choose K- The potential number of neighbors	Parameter
Choose d- fraction of anomalies in data … eg. d=0.01 (1%)	Parameter
For each point in the dataset, find the K nearest neighbors in data	
Compute if the density is high enough	

Choose more variables, which increases the dimension
If the density of a point is much smaller than the densities of its neighbors (LOF ≫1), the point is far from dense areas and, hence, an outlier.
K = 1 emphasizes local anomalies within the data, even if they are not at the boundary; it is more erroneous when having much noise in the data.
Even if point 'a' is not an outlier in x and y axis, it may be one in LOF
Large K can miss local outliers; LOF=1 → no outlier; LOF >> 1 → outlier