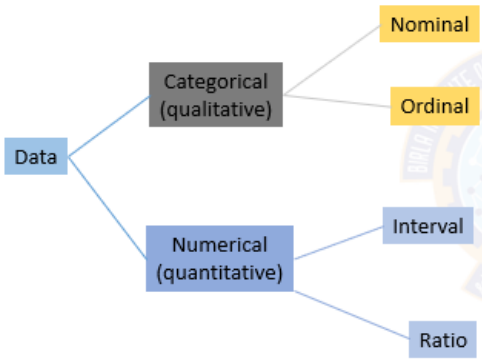
Four Levels of data



- Used for names or label a series of values.

  Ex:Gender(dichotomous),Hair color, Country names

- Ordinal values represent discrete and ordered units.

  Ex: Shirt size, letter grades, food quality

- Interval values represent ordered units that have the same difference.

  Ex: temperature of a given place

- Ratio values represent ordered units that have the same difference. There's a value that means "nothing/none."

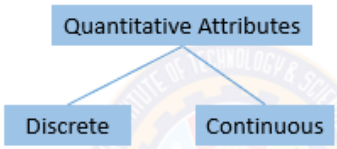
  Ex: Age, weight, test score

Table 2.2. Different attribute types.

Attribute Type		Description	Examples	Operations
Categorical (Qualitative)	Nominal	The values of a nominal attribute are just different names; i.e., nominal values provide only enough information to distinguish one object from another. $(=, \neq)$	zip codes, employee ID numbers, eye color, gender	mode, entropy, contingency correlation, χ^2 test
	Ordinal	The values of an ordinal attribute provide enough information to order objects. $(<, >)$	hardness of minerals, $\{good, better, best\}$, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Numeric (Quantitative)	Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. $(+, -)$	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviatio Pearson's correlation, t and F tests
	Ratio	For ratio variables, both differences and ratios are meaningful. $(*, /)$	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

We can perform more operations as we move from top to bottom.

Flavors for Quantitative Attributes



- Has a finite or countably infinite set of values
- Often integer values are used to denote the values
- Ex: No of employees, set of words in a document collection

- Has real numbers as attribute values
- Often real numbers are used to denote the values
- Ex: height, Weight, Blood sugar levels

Attribute Values

Nominal	Ordinal	Interval	Ratio	Operations
	✓	✓	✓	Ordering
✓	✓	✓	✓	Counts
✓	✓	✓	✓	Mode
	✓	✓	✓	Median
		✓	✓	Mean
		✓	✓	Difference
		✓	✓	Add/Subtract
			✓	Multiply/ Divide
			✓	True Zero

Important characteristics of data

- Dimensionality (number of attributes)
- Sparsity
- Resolution
- Size

Sparsity means whether data has more zeroes. Resolution means at what level we want to capture data like at street level, city level etc.

3. Drop the columns which contain a lot of null or missing values

4. Impute missing values

5. Replace invalid values

6. Remove outliers

7. Log Transform Skewed Variables

8. Transform categorical variables to dummy variables

10. Binning the continuous numeric variables

11. Standardization and Normalization

Data Cleaning

Data Quality issues

- Missing values
- Duplicate data
- Inconsistent / Invalid data
- Noise
- Outliers

Inconsistent / Invalid data

- **Impossible value for a feature**
  - ✓ **Ex: age -10**
  - ✓ **7 letter Income -10000**
  - ✓ **zip code in India**
- **Primarily occur due to data entry error**

Noise

- **It is meaningless or distorted data.**

Customer Name	Address
Shravya	Flat no 450, Street no – 2 Celebrity homes,Mumbai
Shravya	Flat no ęħŦнЖ, Street no – 215, Lave view homes, Bangalore
Mehul	Plot no 80, APARNA SAROVAR ZENITH, Nallagandla, Gachibowli
Shivă	Plot no 80, APARNA SAROVAR ZENITH, Nallagandla, Gachibowli

Data Preprocessing techniques

- Feature selection
  - ✓Adding or removing features
- Feature Transformation
  - ✓Scaling
  - ✓Dimension reduction

Impact of Missing data

- Incompatible in Scikitlearn
- Missing data imputation may distort variable distribution
- Affect the performance of Machine Learning Models

Missing Data : Mechanism

- Understanding the mechanism of missing data will help us choose appropriate imputation method.
  - ✓Missing completely at random (MCAR)
  - ✓Missing at random (MAR)
  - ✓Not missing at random (NMAR):

Missing completely at random (MCAR)

- The probability of missing is same for all the observations.
- There is no relationship between the missing values and any other values in the dataset.
- Removing such missing values will not effect the inferences made.

Missing Data at Random(MAR)

- The **probability of a missing values depends on available information** i.e it depends on other variables in the dataset.

Gender	Age
Male	42
Male	NA
Male	24
Male	NA
Male	36
Male	57
Female	32
Female	NA
Female	NA
Female	18
Female	NA
Female	23

33% males

50% Females

Missing Data Not at Random(MNAR)

- The missing values exist as an indication of a certain class.

No of clinical visits	No of sports classes attended	Depression
1	NA	Yes
NA	NA	Yes
NA	0	Yes
4	2	Yes
NA	1	Yes
3	NA	Yes
0	0	No
NA	5	No
1	2	No
1	1	No
2	1	No
NA	2	No

Imputation Techniques for numeric values

- Mean / Median Imputation
- Random Sampling Imputation
- Adding a new variable to indicate missingness
- Imputation of NA by values at the end of distribution
- Imputation of NA by arbitrary values

Imputation Techniques for categorical values

- Imputation by most frequent category
- In categorical variables treating NA as an additional category

Duplicate Data

- Delete old data
- Merge duplicate records

Invalid Data

- Use external knowledge bases to get the right values
- Apply reasoning and domain knowledge to come with a reasonable estimate

Noise

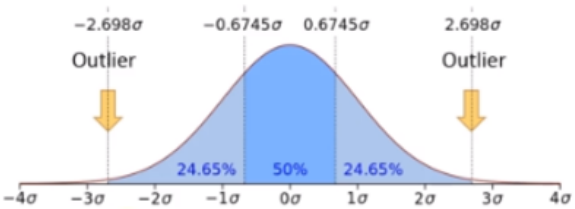
- Filter out the noise component
- This may result in partial loss of data if not done carefully.

Outliers

- Algorithms like Linear Regression, K-Nearest Neighbor, Adaboost are sensitive to noise.
- Outlier can significantly skew the distribution of your data.
- Outliers can be identified using summary statistics and plots of the data.

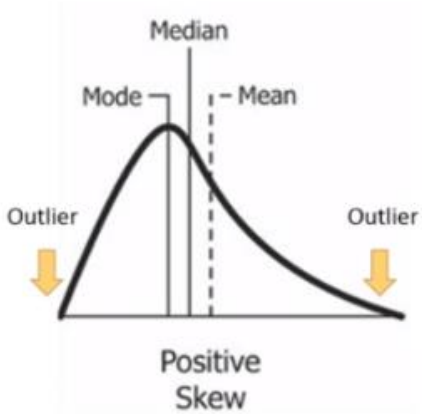
Outliers (cond..)

- Detecting outliers using Normal distribution



- 99% of the observations of a variable following a normal distribution lie within mean +/- 3 X standard deviation

Outliers (cond..)



- Calculate the quantiles and the Inter-quantile range(IQR)

IQR = 75th Quantile – 25th Quantile

Upperlimit = 75th Quantile + IQR x 1.5

Lowerlimit = 25th Quantile + IQR x 1.5

Anomaly detection (aka outlier analysis) is a step in data mining that identifies data points, events, and/or observations that deviate from a dataset's normal behavior. **Anomalous** data

can indicate critical incidents, such as a technical glitch, or potential opportunities, for instance a change in consumer behavior.

Imputation Techniques for Numeric values

Mean / Median Imputation

- Used when MCAR / MAR
- Assumes that the feature follows normal distribution

- Mean age = 33.14
- Median age = 32

Gender	Age
Male	42
Male	NA
Male	24
Male	NA
Male	36
Male	57
Female	32
Female	NA
Female	NA
Female	18
Female	NA
Female	23

Pros and cons of mean/median imputation

- Advantages
  - ✓Easy to implement
  - ✓Faster way of obtaining complete dataset

- Disadvantages
  - ✓Mean imputation reduces the variance of the imputed variables.
  - ✓Mean imputation does not preserve relationships between variables such as correlations.

.

Random Sampling Imputation

- Used when MCAR / MAR
- Aim to preserve the statistical parameters of the feature
- Number of random samples are at least as many as missing values

Gender	Age
Male	42
Male	NA
Male	24
Male	NA
Male	36
Male	57
Female	32
Female	NA
Female	NA
Female	18
Female	NA
Female	23

Adding a new variable to indicate missingness

- Used when MCAR / MAR

Gender	Age	Age Missing?
Male	42	0
Male	NA	1
Male	24	0
Male	NA	1
Male	36	0
Male	57	0
Female	32	0

Imputation of NA by arbitrary values

- Used when NMAR
- Use any value except mean/median value

Gender	Age
Male	42
Male	NA
Male	24
Male	NA
Male	36
Male	57
Female	32

Imputation by most frequent category for categorical values

- Used when NMAR
- Mode = Male

Gender	Age
Male	42
Male	43
Male	24
NA	63
Male	36
Male	57
Female	32
NA	33
Female	45

In categorical variables treating NA as an additional category

- Encode as unique category as unknown or missing
- Use mode to fill missing value

Gender	Gender_new	Gender_new_value
Male	Male	Male
Male	Male	Male
Male	Male	Male
NA	Missing	Male
Male	Male	Male
Male	Male	Male
Female	Female	Female

Aggregation

- Combining two or more attributes (or objects) into a single attribute (or object)

- Purpose
  - ✓Data reduction
  - ✓Change of scale
  - ✓More stable data

Max temp, min temp => max – min

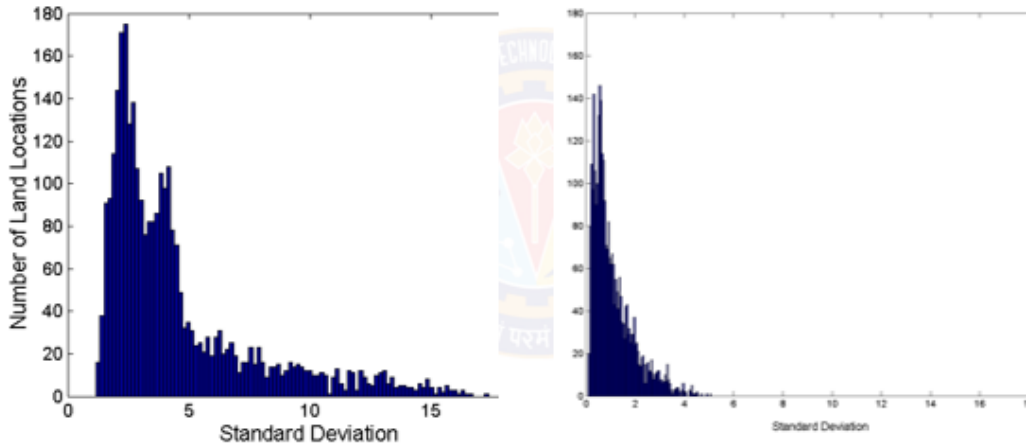
Height, weight => BMI

Male, female => group of male, group of female

When we have lesser aggregate data, it is easy to capture the details well.

Aggregation – Change in Variability

Less variability at "higher-level" view



Standard Deviation of Average Monthly Precipitation

Standard Deviation of Average Yearly Precipitation

Here, we can see the peaks easily in annual graph.

Types of Sampling

- Simple Random Sampling
- Stratified sampling

Hard to work with whole data, use sampling.

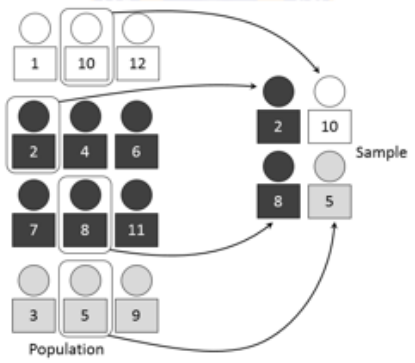
Simple Random Sampling

- **Sampling without replacement:** As each item is selected, it is removed from the population
- **Sampling with replacement:** Objects are not removed from the population as they are selected for the sample.
  - ✓In sampling with replacement, the same object can be picked up more than once.

Use sampling with replacement when the dataset is small.

Stratified sampling

- Split the data into several partitions, then draw random samples from each partition



Use this when we have more males and fewer females. Get the equal no. of representations from each group.

Feature Creation

- Create new attributes that can capture important information in a data set much more efficiently than the original attributes.

- Three general methodologies:
  - Feature Extraction
  - Mapping Data to New Space
  - Feature Construction

Mapping Data to a New Space

- Fourier transform
- Wavelet transform
- Scale-Invariant Feature Transform (SIFT)

In video, images etc, low level features(pitch for audio), mid level features(using fourier), high level features(tags like genre) are available.

Feature construction

- **Create dummy features: Often used to convert categorical variable into numerical variables.**

Customer_ID	Gender	Paymet_Method	Online Banking	Credit Card	Debit Card
C001	FEMALE	Online Banking	1	0	0
C002	MALE	Online Banking	1	0	0
C003	FEMALE	Credit Card	0	1	0
C004	MALE	Debit Card	0	0	1

Needed for neural networks as it expects numeric data alone

Feature construction

- **Create derived features**

Customer_ID	Gender	Session_Begin	Session_End	Session_Duration
C001	FEMALE	15-06-2019 10.30	15-06-2019 11.15	45
C002	MALE	13-06-2019 8.00	13-06-2019 8.03	3
C003	FEMALE	2-06-2019 16.25	2-06-2019 18.35	126
C004	MALE	1-06-2019 11.20	1-06-2019 1.00	100

Derived features examples

- **Commonly used tricks**
  - ✓Handling date, time and addresses
  - ✓Handling sales and Marketing data
  - ✓Handling large range of values using standard transformations
  - ✓Encoding special objects as influencing person

Sometimes the dataset has minimal info. Get additional attributes from external world like distance from hospitals etc.

https://www.analyticsvidhya.com/blog/2013/11/simple-manipulations-extract-data/

Learning Objectives

- List and define various data transformation methods.
- Articulate the need for feature scaling.
- Make the calculations that are necessary to get meaningful transformations.

Data Transformation

- **Data transformation tasks:**
  - ✓Normalization
  - ✓Attribute construction
  - ✓Aggregation
  - ✓Attribute Subset Selection
  - ✓Discretization
  - ✓Generalization

Discretization is the process of converting variables to discrete variables via binning etc.

Generalization is about splitting the training and test data

Linear Models

- $Y = W_0 + W_1 X$
- W indicates the change in Y per unit change of X
- If X changes scale, W will change its value
- Regression coefficients depend on the magnitude of the variable
- Features with bigger magnitude dominate over the features with smaller magnitudes
- Euclidian distances are sensitive to feature magnitude
- Hence it is a good practice to have all variables within a similar scale.

If X is a distance variable and in kilometers and we change it into miles

Gradient descent converges faster when all the variables are in the similar scale

Feature scaling helps decrease the time of finding support vectors

Euclidian distances are sensitive to feature magnitude

For distance-based methods, normalization helps prevent attributes with initially large ranges (e.g., income) from out-weighing attributes with initially smaller ranges (e.g., binary attributes).

Algorithms that are sensitive to feature magnitude

- Linear and Logistic Regression
- Neural Networks
- Support Vector Machines
- KNN
- K-Means Clustering
- Linear Discriminant Analysis (LDA)
- Principal Component Analysis (PCA)

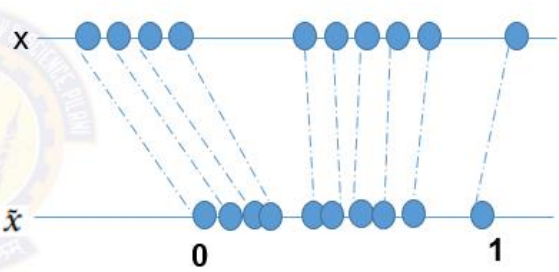
Normalization

- Types of common scaling operations or Normalization methods
  - ✓Min-max normalization
  - ✓z-score normalization
  - ✓Normalization by decimal scaling

Min-Max Scaling

- Min-max scaling squeezes (or stretches) all feature values to be within the range of [0, 1].

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)}$$



x - feature value

min(x) - minimum value of feature x

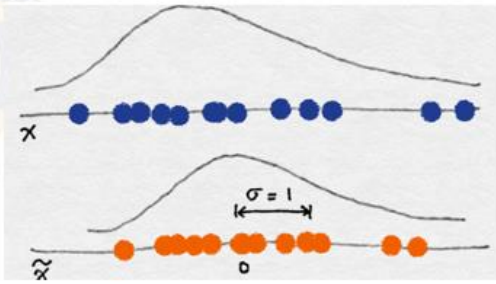
max(x) - maximum value of feature x

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)}(new_max - new_min) + new_min$$

z-score normalization

- In z-score normalization (or zero-mean normalization)
- The values for an attribute, x, are normalized based on the mean (x) and standard deviation (σ_x) of x.

$$\tilde{x} = \frac{x - \mathrm{mean}(x)}{\mathrm{sqrt}(\mathrm{var}(x))}$$



- The resulting scaled feature has a mean of 0 and a variance of 1.

Example: z-score Normalization

Let μ_x= 54,000, σ_x = 16,000, for the attribute income

With z-score normalization, a value of $73,600 for income is transformed to:

$$\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$$

Z=score normalization can change the original data quite a bit.

Decimal Scaling

- Normalizes by moving the decimal point of values of attribute A.
- The number of decimal points moved depends on the maximum absolute value of A.
- A value, v, of A is normalized to v′ by computing

$$V' = \frac{V}{10^j}$$

where j is the smallest integer such that Max(|v′|) < 1.

Examples of Decimal Scaling

Example-1

CGPA	Formula	Normalized CGPA
2	2/10	0.2
3	3/10	0.3

Example-2

Bonus	Formula	Normalized Bonus
400	4/1000	0.4
310	3/1000	0.31

Example-3

Salary	Formula	Normalized Salary
40000	4/100000	0.4
31000	3/100000	0.31

Transformations:

1. **Create variables for difference in date, time and addresses**

While you might be using date and time values on their own, you can create new variables by considering differences in dates and time. Here is an example hypothesis: An applicant who takes days to fill in an application form is likely to be less interested / motivated in the product compared to some one who fills in the same application with in 30 minutes. Similarly, for a bank, time elapsed between dispatch of login details for Online portal and customer logging in might show customers' willingness to use Online portal.
Another example is that a customer living closer to a bank branch is more likely to have a higher engagement than a customer living far off.

2. Create new ratios and proportions

Instead of just keeping past inputs and outputs in your dataset, creating ratios out of them might add a lot of value. Some of the ratios, I have used in past are: Input / Output (past performance), productivity, efficiency and percentages. For example, in order to predict future performance of credit card sales of a branch, ratios like credit card sales / Sales person or Credit card Sales / Marketing spend would be more powerful than just using absolute number of card sold in the branch

3.Apply standard transformations

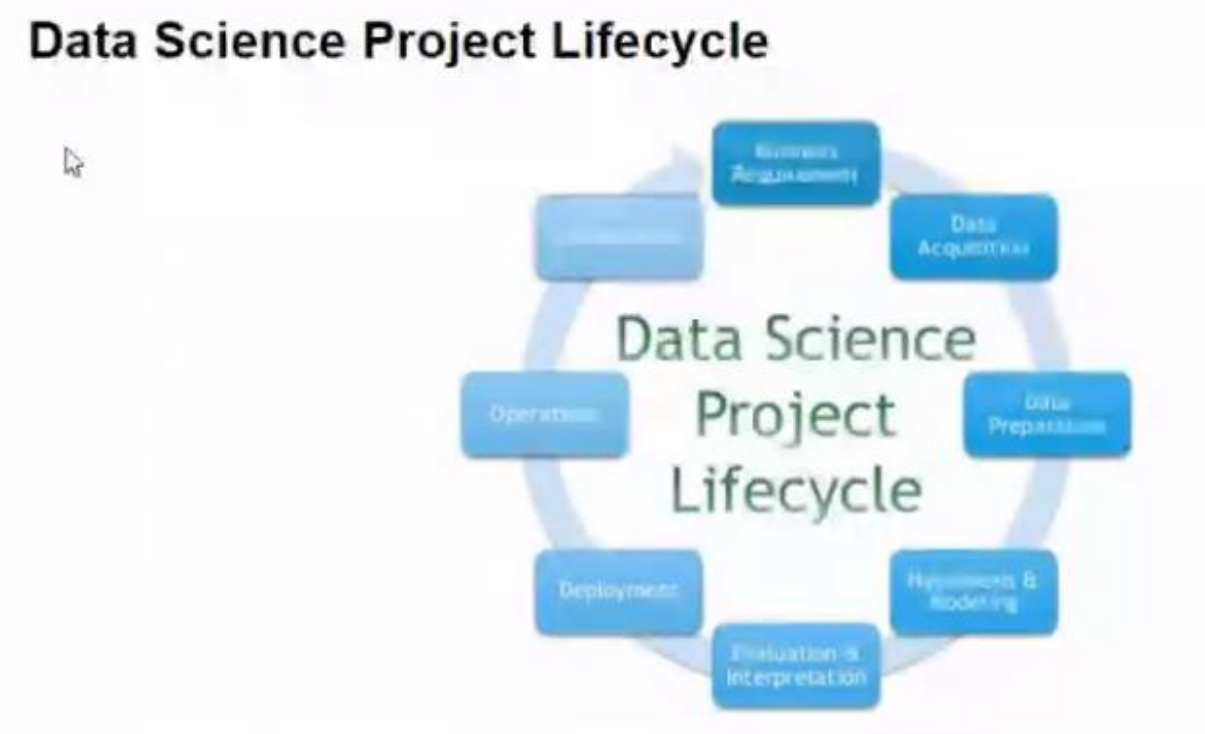
By looking at variations and plots of a variable along with output, you can see if applying basic transformations on variables creates a better relationship. Most commonly used transformations include Log, exponential, quadratic and trignometric variations. For example, Log(Marketing spend) might have a more representable relationship with Sales as compared to absolute Marketing spend.

4. Include effect of influencer

Influencer can impact on the behaviour of your study significantly. Influencer could be of various form and sizes. It could be an employee of your Organization, agent of your Organization or a customer of your Organization. Bringing the impact of these related entities can improve the models significantly. For example, a loan initiated by a sub-set of brokers (and not all brokers) might be more likely to be transferred to a different entity after the lock-in period. Similarly, there might be a sub set of Sales personnel involved who do a higher cross-sell to their customers.

5. Check variables for seasonality and create the model for right

A lot of businesses face some kind of seasonality. It could be driven by tax benefits, festive season or weather. If this is the case, you need to make sure that the data and variables are chosen for the right period. This article can give you more details on how to tackle effects of seasonality while modeling.



Module 2:

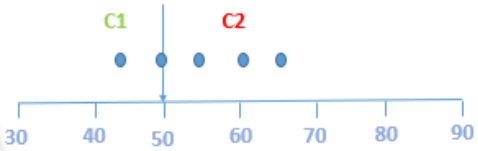
Discretization: (Binning)

Converting continuous variables into discrete variables to work with algorithms like decision trees that expect only discrete variables.

Discretization

➢ **Convert continuous attribute into a discrete one**

Age	Alzimer
60	Yes
65	Yes
45	No
55	Yes
50	No



Main issues
- ✓ How to choose the number of intervals K ?
- ✓ How to define the cut points ?
- ✓ … which are relevant according to the studied problem….

Discretization

➢Unsupervised discretization
- ✓ Equal-interval binning
- ✓ Equal-frequency binning

- Class labels are ignored
- The best number of bins k is determined experimentally

➢Supervised discretization
- ✓Entropy-based discretization
- ✓It tries to maximize the "purity" of the intervals (i.e. to contain as less as possible mixture of class labels)

Unsupervised Discretization

- User specifies the number of intervals and/or how many data points should be included in any given interval.
- The following heuristic is often used to choose intervals:
  - ✓The number of intervals for each attribute should not be smaller than the number of classes (if known).
  - ✓The other popular heuristic is to choose the number of intervals, n_{Fi}, for each attribute, F_i (i=1,…,n,) where n is the number of attributes), as follows:

    n_{Fi} = M/3∗ C where M is the number of training examples and C is the number of known classes.

Methods for Binning Numeric Predictor Variables

- Equal width binning
- Equal frequency binning
- Binning by clustering

Original data:	53 56 57 63 66 67 67 67 68 69 70 70 70 70 72 73 75 75 76 76 78 79 80 81			
Method		**Bin1**	**Bin2**	**Bin3**
Equi Width	81-53=28 28/3=9.33	**[53,62)= {53,56,57}**	**[62,72)= {63,66,67,67,67,68, 69,70,70,70}**	**[72,81]={72,73,75,75,76, 76,78,79,80,81}**
Equi Frequency	24/3=8	**{53,56,57,63,66,67, 67,67}**	**{68,69,70,70,70, 72,73,75}**	**{75,76,76,78,79,80,81}**
Find natural gaps in the data	some variation	**{53,56,57,63,66,67, 67,67,68,69}**	**{70,70,70, 72,73,75,75}**	**{76,76,78,79,80,81}**

The above 3 are unsupervised binning techniques.

Equi width is sensitive to outliers.

Equi frequency can have duplicate values across different bins

Last one is done using clustering to identify natural groups.

Binning using supervised learning:

In this case, we use entropy to measure the right binning for a given variable.

Entropy is the measure of how pure a sample or bin is.

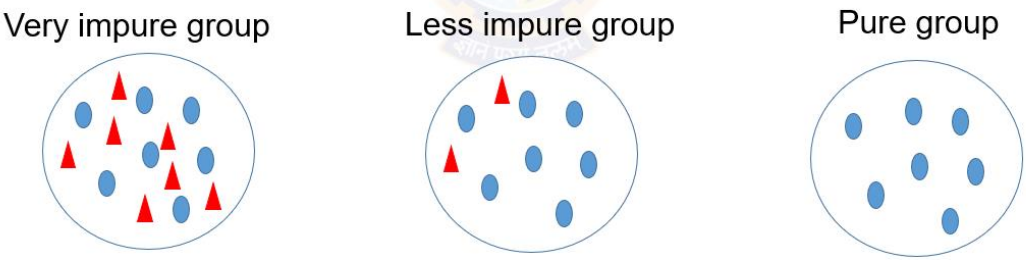
If entropy is very high, then the sample is impure. If the sample contains only one class, then it is the pure sample and entropy will be 0.

How to calculate entropy:

Entropy

- **Entropy E(S) - measure of the impurity/uncertanity in a group of examples**
  - ✓S - training set with $C_1,\ldots,C_C$ classes
  - ✓p_i - proportion of C_c in S

$$E(S) = \sum_{i=1}^{c} -pilog_2p_i$$



Runs	53	56	57	63	66	67	67	67	68	69	70	70	70	70	72	73	75	75	76	76	78	79	80	81
Matches Won	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	N	N	N	Y	N	N	N	N	N	N

Entropy for matches won:

Matches Won	
Y	N
7	17

7/24 = 0.29, 17/24 = 0.71

E(7, 17) = E(0.29, .71)

= -0.29 x $\log_2$(0.29) - 0.71 x $\log_2$(0.71)

= 0.871

Here, higher value of E shows that there is impurity in the group

Information Gain

- Information gain (IG) measures how much "information" a feature gives us about the class.
  - ✓ Features that perfectly partition should give maximal information.
  - ✓ Unrelated features should give no information.
- It measures the reduction in entropy.

$$E(S,A) = \sum_{v \in A} \frac{|S_v|}{|S|} E(S_v)$$

Where S is the number of samples in the training set, with S_v instances belonging to class i, where i = 1, … , c.

Example:

Supervised Discretization Example

Runs	53	56	57	63	66	67	67	67	68	69	70	70	70	70	72	73	75	75	76	76	78	79	80	81
Matches Won	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	N	N	N	Y	N	N	N	N	N	N

In this example we are discretizing the feature Runs using 2 bins <=60 and >60.

Step 3: Calculate "Entropy" for the target.

Matches Won	
Y	N
7	17

E (Runs) = E(7, 17) = E(0.29, .71)
= -0.29 x $\log_2$(0.29) - 0.71 x $\log_2$(0.71)
= 0.871

Step 4: Calculate "Entropy" for the target given a bin.

		Matches Won	
		Y	N
Runs	<= 60	3	0
	> 60	4	17

E (Matches Won,Runs) = P(<=60) x E(3,0) + P(>60) x E(4,17)
= 3/24 x 0 + 21/24 x 0.7= 0.615

Step 5: Calculate "Information Gain" given a bin.
Information Gain (Matches Won,Runs) = 0.256

Here p(<=60) is the proportion of <60 in the total set ie., 3/24…

Here .256 is obtained as .871 - .615

Supervised Discretization - Entropy Based

- **Entropy Based Discretization**

1. Sort examples in increasing order
2. Each value forms an interval (m intervals)
3. Calculate the entropy measure of this discretization $E(S) = \sum_{i=1}^{c} -p_i log_2 p_i$
4. Calculate "Entropy" for the target given a bin. $E(S,A) = \sum_{v \in A} \frac{|S_v|}{|S|} E(S_v)$
5. Calculate "Information Gain" given a bin. E(S) - E(S,A)
6. Apply the process recursively until some stopping criterion is met.

$$E(S) - E(S,A) > \delta$$

Supervised Discretization Example (Contd..)

Runs	53	56	57	63	66	67	67	67	68	69	70	70	70	70	72	73	75	75	76	76	78	79	80	81
Matches Won	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	Y	N	N	N	Y	N	N	N	N	N	N

		Matches Won	
		Y	N
Runs	<= 60	3	0
	> 60	4	17

Information Gain = 0.256

		Matches Won	
		Y	N
Runs	<= 70	6	8
	> 70	1	9

Information Gain = 0.101

		Matches Won	
		Y	N
Runs	<= 75	7	11
	> 75	0	6

Information Gain = 0.148

Based on above example, we know that the first split gives more information gain, so it is the right binning strategy.

Binarization

- Binarization maps a continuous or categorical attribute into one or more binary variables.
- Must maintain ordinal relationship

✓Assume an ordinal attribute for representing service of a restaurant: ({Awful,Poor,OK,Good,Great})

Service quality	Integer Value	X1	X2	X3
Awful	0	0	0	0
Poor	1	0	0	1
Ok	2	0	1	0
Good	3	0	1	1
Great	4	1	0	0

Unintended relationships: X2 and X3 are now correlated because "good" is encoded using both attributes

Here, there is high correlation between X2 and X3. This needs to be avoided. Use

One hot encoding:

Binarization

Service quality	Integer Value	X1	X2	X3	X4	X5
Awful	0	1	0	0	0	0
Poor	1	0	1	0	0	0
Ok	2	0	0	1	0	0
Good	3	0	0	0	1	0
Great	4	0	0	0	0	1

✓Binary attributes, where only the presence of 1 is important

✓One binary attribute for each categorical value

✓Be Careful: Number of resulting attributes may become too large

Here, the no. of features is more.

Proximity:

Proximity

- For many problems we need to quantify how close two objects are.
- Examples:
  - ✓For an item bought by a customer, find other similar items
  - ✓Group together the customers of site so that similar customers are shown the same ad.
  - ✓Group together web documents so that you can separate the ones that talk about politics and the ones that talk about sports.
  - ✓Find all the near-duplicate mirrored web documents.
  - ✓Find credit card transactions that are very different from previous transactions.
- To solve these problems we need a definition of similarity, or distance.

Proximity (contd..)

- Similarity
  - ✓Numerical measure of how **alike** two data objects are.
  - ✓Is higher when objects are more alike.
  - ✓Often falls in the range [0,1]
  - ✓Examples: Cosine,Jaccard,Tanimoto
- Dissimilarity
  - ✓Numerical measure of how **different** two data objects are
  - ✓Lower when objects are more alike
  - ✓Minimum dissimilarity is often 0
  - ✓Upper limit varies

Proximity Measures for Single Nominal attribute

✓ Suppose a binary attribute Gender = {Male, female} where Male is equivalent to binary 1 and female is quivalent to binary 0.

✓ The similarity value(p) is 1 if the two objects contains the same attribute value, 0 otherwise.

Object	Gender
Ram	Male
Sita	Female
Laxman	Male

✓ $p(Ram, sita) = 0$

✓ $p(Ram, Laxman) = 1$

✓ Note : In this case, if q denotes the dissimilarity between two objects i and j with single binary attributes, then $q_{(i,j)} = 1 - p_{(i,j)}$

Proximity Measures for Two or more Nominal attribute

✓We define the contingency table summarizing the different matches and mismatches between any two objects x and y, which are as follows.

Contingency table with binary attributes

Object x	Object y: 1	Object y: 0
1	f_{11}	f_{10}
0	f_{01}	f_{00}

Here, f_{11}= the number of attributes where x=1 and y=1.

f_{10}= the number of attributes where x=1 and y=0.

f_{01}= the number of attributes where x=0 and y=1.

f_{00}= the number of attributes where x=0 and y=0

✓Note : $f_{00} + f_{01} + f_{10} + f_{11}$ will total number of binary attributes.

✓Two cases of binary attributes may arise: symmetric and asymmetric binary attributes.

For symmetric variables, the importance is same for all possible outcomes.

For asymmetric variables, one outcome is more important than other when measuring proximity. For ex, if two patients undergo a test, a positive result will be more important to measure if two patients are similar or not.

For asymmetric, we use Jaccard coefficient.

Similarity Measure for Symmetric Binary attribute

- Symmetric binary coefficient($\mathcal{S}$) is used to measure the similarity between two objects and is defined as

$$\mathcal{S} = \frac{Number\ of\ matching\ attribute\ values}{Total\ number\ of\ attributes}$$

or

$$\mathcal{S} = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

- The dissimilarity measure($\mathcal{D}$) is defined as

$$\mathcal{D} = \frac{Number\ of\ mismatched\ attribute\ values}{Total\ number\ of\ attributes}$$

or

$$\mathcal{D} = \frac{f_{01} + f_{10}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

Similarity Measure with Symmetric Binary

Consider the following two dataset, where objects are defined with symmetric binary attributes.

Gender = {M, F}, Food = {V, N}, Caste = {H, M}, Education = {L, I}, Hobby = {T, C}, Job = {Y, N}

Object	Gender	Food	Caste	Education	Hobby	Job
Hari	M	V	M	L	C	N
Ram	M	N	M	I	T	N
Tomi	F	N	H	L	C	Y

$$\mathcal{S}(\text{Hari, Ram}) = \frac{2+1}{2+2+1+1} = 0.5$$

	1	0
1	1	1
0	2	2

To populate this table, consider M as 1 and F as 0, V as 1 and N as 0 etc.

To

Proximity Measure with Asymmetric Binary

- Jaccard Coefficient is used to measure the similarity between two objects is symbolized by $\mathcal{J}$ and is defined as follows

$$\mathcal{J} = \frac{Number\ of\ matching\ presence}{Number\ of\ attributes\ not\ involved\ in\ 00\ matching}$$

or

$$\mathcal{J} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

Proximity Measure with Asymmetric Binary

Consider the following two dataset.

Gender = {M, F}, Food = {V, N}, Caste = {H, M}, Education = {L, I},
Hobby = {T, C}, Job = {Y, N}

Compute the Jaccard coefficient between Ram and Hari assuming that all binary attributes are asymmetric and for each pair values for an attribute, first one is more important than the second.

Object	Gender	Food	Caste	Education	Hobby	Job
Hari	M	V	M	L	C	N
Ram	M	N	M	I	T	N
Tomi	F	N	H	L	C	Y

$$\mathcal{J}(\text{Hari, Ram}) = \frac{1}{2+1+1} = 0.25$$

	1	0
1	1	1
0	2	2

Note: $\mathcal{J}$(Hari, Ram) = $\mathcal{J}$(Ram, Hari)

For nonbinary variables, use following approach

Proximity Measures for Categorical Attribute

- Attributes with three or more states (e.g. color = {Red, Green, Blue}) are called nominal.
- If $s(x, y)$ denotes the similarity between two objects x and y, then

$$s(x,y)=\frac{Number\ of\ matches}{Total\ number\ of\ attributes}$$

- and the dissimilarity $d(x, y)$ is

$$d(x,y)=\frac{Number\ of\ mismatches}{Total\ number\ of\ attributes}$$

- If m = number of matches and a = number of the categorical attribute for object x and y then s and D are defined as

$$s(x,y)=\frac{m}{a} \quad \text{and} \quad d(x,y)=\frac{a-m}{a}$$

Proximity Measures for Categorical Attribute

Object	Color	Position	Distance
1	R	L	L
2	B	C	M
3	G	R	M
4	R	L	H

$$s = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & & & \\ 2 & 0 & 0 & & \\ 3 & 0 & 1/3 & 0 & \\ 4 & 2/3 & 0 & 0 & 0 \end{array}$$

s(1,2)
s(2,1)

$$d = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & & & \\ 2 & 3/3 & 1 & & \\ 3 & 3/3 & 2/3 & 1 & \\ 4 & 1/3 & & & \end{array}$$

For ordinal attributes, the approach is going to be different.

Proximity Measure for Ordinal Attribute

- Ordinal attribute is a special kind of categorical attribute, where the values of attribute follows a sequence (ordering) e.g. Grade = {Ex, A, B, C} where Ex > A >B >C.
- Suppose, *A* is an attribute of type ordinal and the set of values of $A = \{a_1, a_2, \ldots\ldots, a_n\}$. Let n values of A are ordered in ascending order as $a_1 < a_2 < .. < a_n$. Let *i-th* attribute value a_i be ranked as *i, i=1,2,..n.*
- The normalized value of a_i can be expressed as

$$\hat{a}_i = \frac{i-1}{n-1}$$

- Thus, normalized values lie in the range $[0..1]$.
- As a_i is a numerical value, the similarity measure, then can be calculated using any similarity measurement method for numerical attribute.
- For example, the similarity measure between two objects x *and* y with attribute values a_i and a_j, then can be expressed as

$$s(x,y) = \sqrt{(\hat{a}_i - \hat{a}_j)^2}$$

where $\hat{a}_i$ and $\hat{a}_i$ are the normalized values of $\hat{a}_i$ and $\hat{a}_i$, respectively.

Proximity Measure for Ordinal Attribute

Consider the following set of records, where each record is defined by two ordinal attributes size={S, M, L} and Quality = {Ex, A, B, C} such that S<M<L and Ex>A>B>C.

Object	Size	Quality
A	S (0.0)	A (0.66)
B	L (1.0)	Ex (1.0)
C	L (1.0)	C (0.0)
D	M (0.5)	B (0.33)

S-1=0 $\frac{1-1}{3-1}$
M-2=0.5
L-3=1
A-1=0 $\frac{1-1}{4-1}$
B-2 $\frac{2-1}{4-1}$
C-3 $\frac{3-1}{4-1}$
Ex-4 $\frac{4-1}{4-1}$

$d(A,B) = \sqrt{(0.0-1.0)^2 + (0.66-1.0)^2}$

Find the dissimilarity matrix, when each object is defined by only one ordinal attribute say size (or quality).

Distance between features:

Properties of distance measures

- Distance *d (p, q)* between two points *p* and *q* is a dissimilarity measure if it satisfies:

  1. **Positive definiteness:**

     $d(p, q) \geq 0$ for all *p* and *q* and

     $d(p, q) = 0$ only if $p = q$.

  2. **Symmetry:** $d(p, q) = d(q, p)$ for all *p* and *q*.

  3. **Triangle Inequality:**

     $d(p, r) \leq d(p, q) + d(q, r)$ for all points *p*, *q*, and *r*.

Proximity Measure with Interval Scale

- The generic formula to express distance d between two objects x *and* y in *n*-dimensional space.

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^r \right)^{\frac{1}{r}}$$

Here, r is any integer value, x_i *and* y_i denote the values of ith attribute of the objects x *and* y respectively

- This distance metric most popularly known as Minkowski metric.

Proximity Measure with Interval Scale

Manhattan distance (L_1 Norm: r = 1)

The Manhattan distance is expressed as

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

where $|\ldots|$ denotes the absolute value.

This metric is also alternatively termed as **Taxicabs metric, city-block metric**.

Example: x = [7, 3, 5] and y = [3, 2, 6].

The Manhattan distance is $|7-3| + |3-2| + |5-6| = 6$.

- As a special instance of Manhattan distance, when attribute values $\in [0, 1]$ is called Hamming distance.
- Alternatively, Hamming distance is the number of bits that are different between two objects that have only binary values (i.e. between two binary vectors).

Proximity Measure with Interval Scale

Euclidean Distance (L_2 Norm: r = 2)

This metric is same as Euclidean distance between any two points $x\ and\ y\ in\ \mathcal{R}^n$.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Example: x = [7, 3, 5] and y = [3, 2, 6].

The Euclidean distance between $x\ and\ y$ is

$$d(x, y) = \sqrt{(7-3)^2 + (3-2)^2 + (5-6)^2} = \sqrt{18} \approx 2.426$$

Proximity Measure with Interval Scale

Chebychev Distance ($L_\propto$ Norm: $r \in \mathcal{R}$)

This metric is defined as

$$d(x, y) = \max_{\forall i}\{|x_i - y_i|\}$$

Example: x = [7, 3, 5] and y = [3, 2, 6].

The Manhattan distance = $|7 - 3| + |3 - 2| + |5 - 6| = 6.$

The chebychev distance = $\text{Max}\ \{|7 - 3|, |3 - 2|, |5 - 6|\} = 4.$

Proximity Measure for Ratio scale

The proximity between the objects with ratio-scaled variable can be carried with the following steps:

1. Apply appropriate transformation to the data to bring it into a linear scale. (e.g. logarithmic transformation to data of the form $X = Ae^B$.
2. The transformed values can be treated as interval-scaled values. Any distance measure discussed for interval-scaled variable can be applied to measure the similarity.

Proximity Measure for Ratio scale

Normalization:

- A major problem when using the similarity (or dissimilarity) measures (such as Euclidean distance) is that the large values frequently swamp the small ones.
- For example, consider the following data.

Make	Cost 1	Cost 2	Cost 3
X	2,00,000	70	10
Y	2,50,000	100	5

- Here, the contribution of Cost 2 and Cost 3 is insignificant compared to Cost 1 so far the Euclidean distance is concerned.
- This problem can be avoided if we consider the normalized values of all numerical attributes.

Proximity Measure for Mixed Attributes

- The previous metrics on similarity measures assume that all the attributes were of the same type. Thus, a general approach is needed when the attributes are of different types.
- One straightforward approach is to compute the similarity between each attribute separately and then combine these attribute using a method that results in a similarity between 0 and 1.
- Typically, the overall similarity is defined as the average of all the individual attribute similarities.

Proximity Measure with Vector Objects

Suppose, the objects are defined with $A_1, A_2, \ldots\ldots, A_n$ attributes.

1. For the *k-th* attribute (*k* = 1, 2, . . . , *n*), compute similarity $s_k(x, y)$ in the range [0, 1].
2. Compute the overall similarity between two objects using the following formula

$$similarity\ (x, y) = \frac{\sum_{i=1}^{n} s_i(x, y)}{n}$$

3. The above formula can be modified by weighting the contribution of each attribute. If the weight w_k is for the *k-th* attribute, then

$$w_similarity\ (x, y) = \frac{\sum_{i=1}^{n} w_i s_i(x,y)}{n} \text{ such that } \sum_{i=1}^{n} w_i = 1.$$

4. The definition of the Minkowski distance can also be modified as follows:

$$d(x, y) = \left(\sum_{i=1}^{n} w_i |x_i - y_i|^r \right)^{\frac{1}{r}}$$

Proximity Measure with Mixed Attributes

Consider the following set of objects.

Object	A (Binary)	B (Categorical)	C (Ordinal)	D (Numeric)	E (Numeric)
1	Y	R	X	475	10^8
2	N	R	A	10	10^{-2}
3	N	B	C	1000	10^5
4	Y	G	B	500	10^3
5	Y	B	A	80	1

Here we have mixed type of variables. IF we feel that A variable is more important, give highest weight to it and so on.

Non-Metric Similarity

- In many applications (such as information retrieval) objects are complex and contains a large number of symbolic entities (such as keywords, phrases, etc.).
- To measure the distance between complex objects, it is often desirable to introduce a non-metric similarity function.

Cosine similarity

Suppose, *x* and *y* denote two vectors representing two complex objects. The cosine similarity denoted as $\cos(x, y)$ and defined as

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

- where $x \cdot y$ denotes the vector dot product, namely $x \cdot y = \sum_{i=1}^{n} x_i \cdot y_i$ such that $x = [x_1, x_2, .., x_n]$ and $y = [y_1, y_2, .., y_n]$.
- $\|x\|$ and $\|y\|$ denote the Euclidean norms of vector *x* and *y*, respectively (essentially the length of vectors *x* and *y*), that is
  - $\|x\| = \sqrt{x_1^2 + x_2^2 + .. + x_n^2}$ and $\|y\| = \sqrt{y_1^2 + y_2^2 + .. + y_n^2}$

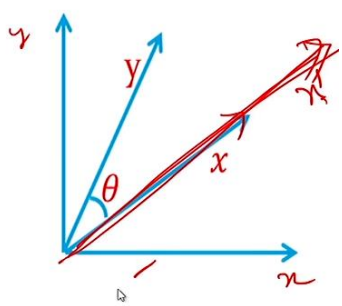
Cosine Similarity

- In fact, cosine similarity essentially is a measure of the (cosine of the) angle between *x* and *y*.
- Thus if the cosine similarity is 1, then the angle between *x* and *y* is 0° and in this case, *x* and *y* are the same except for magnitude.
- On the other hand, if cosine similarity is 0, then the angle between *x* and *y* is 90° and they do not share any terms.
- Considering, this cosine similarity can be written equivalently

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} = \frac{x}{\|x\|} \cdot \frac{y}{\|y\|} = \hat{x} \cdot \hat{y}$$

  where $\hat{x} = \frac{x}{\|x\|}$ and $\hat{y} = \frac{y}{\|y\|}$. This means that cosine similarity does not take the magnitude of the two vectors into account, when computing similarity.

- It is thus, one way normalized measurement.



1, 3, 5

$\frac{1}{\|x\|}, \frac{3}{\|x\|}, \frac{5}{\|x\|}$

By dividing the value with magnitude, there won't be any bias.

Non-Metric Similarity

Cosine Similarity

Suppose, we are given two documents with count of 10 words in each are shown in the form of vectors *x* and *y* as below.

x = [3, 2, 0, 5, 0, 0, 0, 2, 0, 0] and *y* = [1, 0, 0, 0, 0, 0, 0, 1, 0, 2]

Thus, $x \cdot y$ = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5

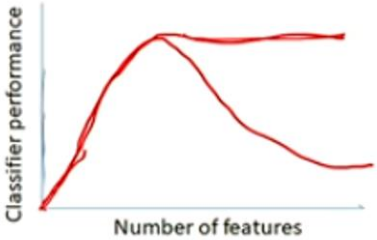
$$\|x\| = \sqrt{3^2 + 2^2 + 0 + 5^2 + 0 + 0 + 0 + 2^2 + 0 + 0} = 6.48$$

$$\|y\| = \sqrt{1^2 + 0 + 0 + 0 + 0 + 0 + 0 + 1^2 + 0 + 2^2} = 2.24$$

$$\therefore \quad \cos(x, y) = 0.31$$

Curse of Dimensionality

- As dimensionality increases the number of data points required for a classification model also increase exponentially.
- Hughes Phenomenon: For a fixed number of training samples(N) in the data set the performance of the models decreases as dimensionality increase.



Reasons for this phenomenon:

- ✓Redundant Features – Carry same data in some other form
- ✓Correlation between features – the presence of one feature influence the other.
- ✓Irrelevant Features - those that are simply unnecessary

Here, after a certain point, the performance is either constant or decreasing.

Irrelevant features could be student id, emp id etc.

Curse of Dimensionality

✓ The intuitions of distances in 3D are invalid in higher dimensions.
✓ For example consider a data point x_i from N samples in 1D

$\text{dist}_{min}(x_i)=\min_{xi \neq xj}\{\text{dist}(x_i,x_j)\}$ The minimum of distance between x_i and x_j such that $x_i \neq x_j$

$\text{dist}_{max}(x_i)=\max_{xi \neq xj}\{\text{dist}(x_i,x_j)\}$ The maximum of distance between x_i and x_j such that $x_i \neq x_j$

$$\lim_{d\to\infty} \frac{\text{dist}_{\text{max}} - \text{dist}_{\text{min}}}{\text{dist}_{\text{min}}} = 0$$

✓ $\text{dist}_{max}(x_i) \approx \text{dist}_{min}(x_i)$ that means every pair of points are approximately at the same distance from each other.

➢ **Distance measures become meaningless in higher dimensions.**

Curse of Dimensionality

✓ The intuitions of distances in 3D are invalid in higher dimensions.
✓ For example consider a data point x_i from N samples in 1D

$\text{dist}_{min}(x_i)=\min_{xi \neq xj}\{\text{dist}(x_i,x_j)\}$ The minimum of distance between x_i and x_j such that $x_i \neq x_j$

$\text{dist}_{max}(x_i)=\max_{xi \neq xj}\{\text{dist}(x_i,x_j)\}$ The maximum of distance between x_i and x_j such that $x_i \neq x_j$

$$\lim_{d\to\infty} \frac{\text{dist}_{\text{max}} - \text{dist}_{\text{min}}}{\text{dist}_{\text{min}}} = 0$$

✓ $\text{dist}_{max}(x_i) \approx \text{dist}_{min}(x_i)$ that means every pair of points are approximately at the same distance from each other.

➢ **Distance measures become meaningless in higher dimensions.**

Euclidean distance VS Cosine similarity

- Euclidean distance in high dimensionality does not make the sense solution for this is using cosine similarity for high dimensional spaces.
- Impact of dimensionality on cosine similarity is lower as compared to the Euclidean distance.
- If the data is dense then it's impact will be high and if it is sparse then impact will be lower that means in sparse most of values are 0 so data is non uniformly spread.

Feature Subset Selection
Dimension Reduction

Importance of feature subset selection

The objective of feature selection is three-fold:

- ✓Improving the prediction performance of the models
- ✓Reduction in the training time required to build model
- ✓Providing a better understanding of the underlying process that generated the data

What is Feature Selection for classification?

- Given: A set of predictors ("features") $F=\{f_1,f_2,f_3\ldots f_D\}$ and target class label T.
- Find: Minimum subset $F'=\{f_1',f_2',f_3'\ldots f_M'\}$ that achieves maximum classification performance where $F' \subseteq F$.

Feature subset selection

- ✓ Given D initial set of features
- ✓ There are 2^D possible subsets.
- ✓ Need a criteria to decide which subset is the best:
  - ✓ Classifier based on these m features has the lowest probability of error of all such classifiers.
- ✓ Evaluating 2^D possible subsets is time consuming and expensive.
- ✓ Use heuristics to reduce the search space.

Feature Selection approaches

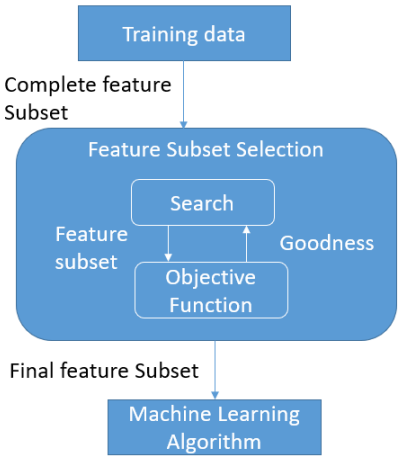
Three approaches to evaluate 2^D possible subsets

- Unsupervised (Filter Methods)
  - ✓ Use only features/predictor variables
  - ✓ Select the features that have the most information
- Supervised: Wrapper Methods
  - ✓ Train using the selected subset
  - ✓ Estimate error on the validation set
- Embedded Methods
  - ✓ Feature selection is done while training the model

Examples of embedded methods: Lasso regression and Decision trees.

Steps in Feature Selection

- Feature selection is an optimization problem having the following steps:
- Step1: Search the space of all possible features
- Step2: Pick the optimal subset using an objective function



The search algorithm will select a feature subset and score it on the objective function and find the goodness of the feature subset and based on this it will decide which part of the sub space to explore next and after this motive is completed we get a final feature subset.

So we want to pick the subset that is optimal with respect to the objective function

Learning Objectives

- Formulate the problem of filter methods for feature subset selection
- List various filters
- Define the Pearson correlation filter for regression
- Explain no free lunch theorem

Filter Methods

- The Predictive power of individual feature is evaluated.
- Rank each feature according to some univariate metric and select the highest ranking features.
- The score should reflect the discriminative power of each feature.

Input: large feature set Ω

1 Identify candidate subset S ⊆ Ω

2 While !stop criterion()

Evaluate utility function J using S.

Adapt S

3 Return S.

Pros: fast, provides generically useful feature set

Cons: cause higher error than wrappers

The utility function J should reflect the discriminative power of each feature.

Cons: A feature that is not useful by itself can be very useful when combined with others. Filter methods can miss it!

Types of Filters

- **Univariate filters** evaluate each feature independently with respect to the target variable.
  - ✓ Correlation
  - ✓ Fisher Score
  - ✓ Mutual Information (Information Gain)
  - ✓ Gini index
  - ✓ Gain Ratio
  - ✓ Chi-Squared test
- **Multivariate filters** evaluate features in context of others.

Types of filters

- Correlation-based
  - ✓ Pearson product-moment correlation
  - ✓ Spearman rank correlation
  - ✓ Kendall concordance
- Statistical/probabilistic independence metrics
  - ✓ Chi-square statistic
  - ✓ F-statistic
  - ✓ Welch's statistic
- Information-theoretic metrics
  - ✓ Mutual Information (Information Gain)
  - ✓ Gain Ratio
- Others
  - ✓ Fisher score
  - ✓ Gini index
  - ✓ Cramer's V

Many variables are correlated with each other and hence are redundant. If two variables are correlated, keeping one will help reduce dimensionality. But which variable to keep and which one to get rid off? Obviously, the one that has higher correlation.

How “useful” is a single feature? : Univariate filters

Trying to predict someone’s ML exam grade from various possible indicators (a.k.a. features):

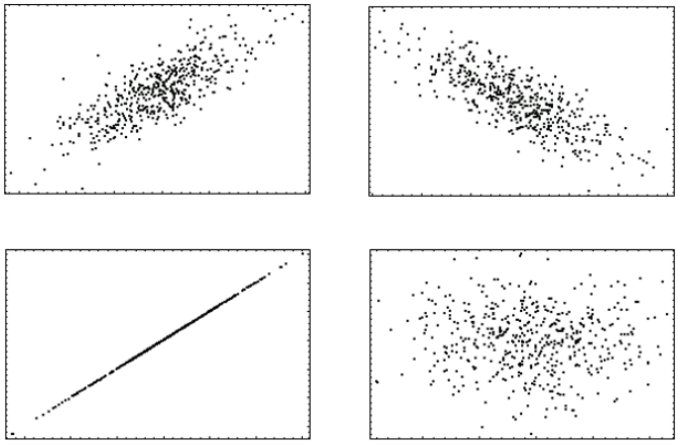
1) Statistics grade,

2) Biology grade,

3) Linear Algebra grade, or

4) Height ...

Which one would you pick?



Many variables are correlated with each other and hence are redundant. If two variables are correlated, keeping one will help reduce dimensionality. But which variable to keep and which one to get rid off? Obviously, the one that has higher correlation.

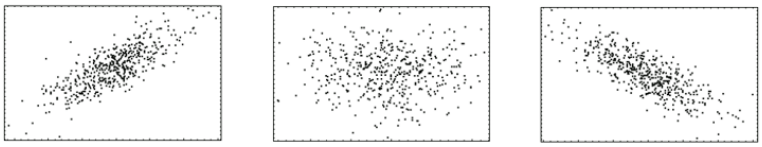
Pearson’s Correlation Coefficient

- Used to measure the strength of association between two continuous random variables.

Feature : $\mathbf{x}_k = \{x_k^{(1)}, ..., x_k^{(N)}\}^T$

Target : $\mathbf{y} = \{y^{(1)}, ..., y^{(N)}\}^T$

$$r(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{N}(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x^{(i)} - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y^{(i)} - \bar{y})^2}}$$



$r = +0.5$ $\quad$ $r = 0.0$ $\quad$ $r = -0.5$

Both positive and negative correlation is useful!

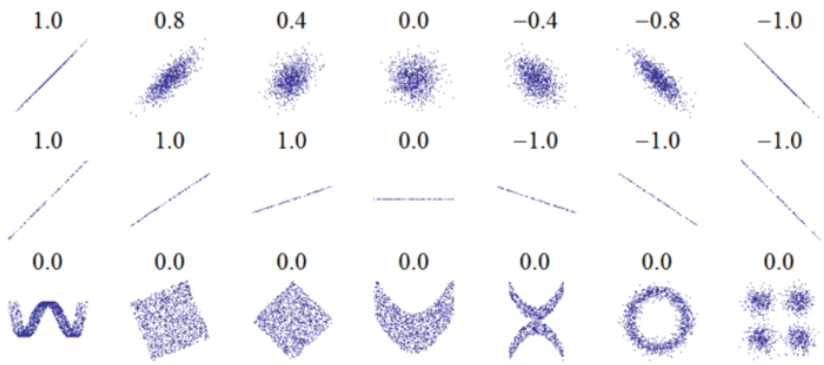
Ranking with Filter Criteria

- Rank features X_i , $\forall i$ by their values of $J(X_k)$.
- Retain the highest ranked features, discard the lowest ranked.

Cut-off point decided by user, e.g. |S| = 5,
S = {35, 42, 10, 654, 22}.

k	$J(X_k)$
35	0.846
42	0.811
10	0.810
654	0.611
22	0.443
59	0.388
...	...
212	0.09
39	0.05

Limitation: Pearson assumes all features are INDEPENDENT ! and... only identifies LINEAR correlations.



There are LOTS of ranking criteria...

Pearson, Fisher, Mutual Info, Jeffreys-Matsusita, Gini Index, AUC, F-measure, Kolmogorov distance, Chi-squared, CFS, Alpha-divergence, Symmetrical Uncertainty,.... etc, etc

- How do I pick the right filter ? Unfortunately, quite complex.... depends on:
  - ✓type of variables/targets (continuous, discrete, categorical).
  - ✓class distribution
  - ✓degree of nonlinearity/feature interaction

- The **"No Free Lunch"** theorem states that there is no universal model that works best for every problem.

Here class distribution assumes that the data follows normal distribution

Many produce very similar rankings....

The No Free Lunch theorem in Machine Learning says that no single machine learning algorithm is universally the best algorithm. In fact, the goal of machine learning models is not find an algorithm that will be the best.

There is no single algorithm which has performed better in all sceneries

Hypothesis Testing

- Hypothesis is a premise or claim that we want to investigate.
- Test whether the two random variables (categorical) are independent or not.

- Test Statistic
  - ✓Chi-Squared Test
  - ✓T-Test
  - ✓ANNOVA-Test

Test Statistic

We will assume that H0 is always true unless the evidence portraits something else.in which case we will reject the Null hypothesis and accept Ha

We are going to test if H0 is true

Mathematically H0 and HA are mathematically opposite of each other

Chi-square test of independence allow us to see whether or not two categorical variables are related or if they are not. If the variables had no effect on each other, then we would expect the distributions across the rows or the columns to be about the same. And the expected count for each cell is the product of the row totals and column totals, divided by the grand total.

Example

A group of customers were classified in terms of personality (introvert, extrovert or normal) and in terms of color preference (red, yellow or green) with the purpose of seeing whether there is an association (relationship) between personality and color preference.

Data was collected from 400 customers and presented in the 3 (rows) x 3 (cols) contingency table below:

(Observed counts)	Colors			
Personality	Red	Yellow	Green	Totals
Introvert personality	11	5	1	17
Extrovert personality	8	6	8	22
Normal	3	10	12	25
Total	22	21	21	64

Five-step approach for Chi-Squared test of independence

- **Step 1.** Set up hypotheses and determine level of significance.
  - ✓Null hypothesis(H0): Color preference is independent of personality.
  - ✓Alternative hypothesis(H_A): Color preference is dependent on personality
  - ✓α=0.05

Five-step approach for Chi-Squared test of independence (contd..)

Step 2. Compute the expected frequency (under the null hypothesis) in each cell using E = (Row Total * Column Total)/N

(Expected counts)	Colors			
Personality	Red	Yellow	Green	Totals
Introvert personality	5.8	5.6	5.6	17
Extrovert personality	7.6	7.2	7.2	22
Normal	8.6	8.2	8.2	25
Total	22	21	21	64

Step 3:Select the test statistic

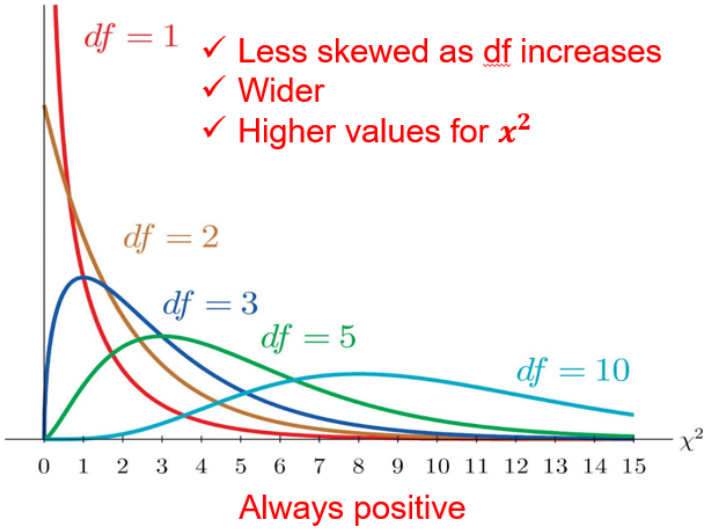
$$x^2 = \Sigma \frac{(0-E)^2}{E}$$

$$x^2 = \frac{(11-5.8)^2}{5.8} + \frac{(5-5.6)^2}{5.6} + \frac{(1-5.6)^2}{5.6} + \ldots + \frac{(12-8.2)^2}{8.2} = 14.5$$

Chi-Squared distribution

The probability density function for the x^2 distribution with r degrees of freedom(df) is given by

$$P_r(x) = \frac{x^{r/2-1}\, e^{-x/2}}{\Gamma\left(\frac{1}{2} r\right) 2^{r/2}}$$



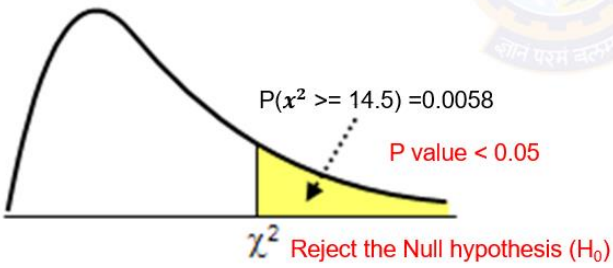
Here you see the distribution with 1,2,3,5 and 10.

The distribution is always positive if you think about it which makes sense since the formula has a square

Five-step approach for Chi-Squared test of independence (Contd..)

- Step 4: Use a probability table to find P-Value associated with x^2 value for with degrees of freedom df = (r − 1) (c − 1), r is the number of categories in one variable and c is the number of categories in the other.

- Step 5: Make a conclusion using P-value



	Significance Level				
df	0.10	0.05	0.025	0.01	0.005
1	2.7055	3.8415	5.0239	6.6349	7.8794
2	4.6052	5.9915	7.3778	9.2104	10.5965
3	6.2514	7.8147	9.3484	11.3449	12.8381
4	7.7794	9.4877	11.1433	13.2767	14.8602
5	9.2363	11.0705	12.8325	15.0863	16.7496
6	10.6446	12.5916	14.4494	16.8119	18.5475
7	12.017	14.0671	16.0128	18.4753	20.2777

https://medium.com/wenyi-yan/a-simple-explanation-to-understand-chi-square-test-1814fa261499

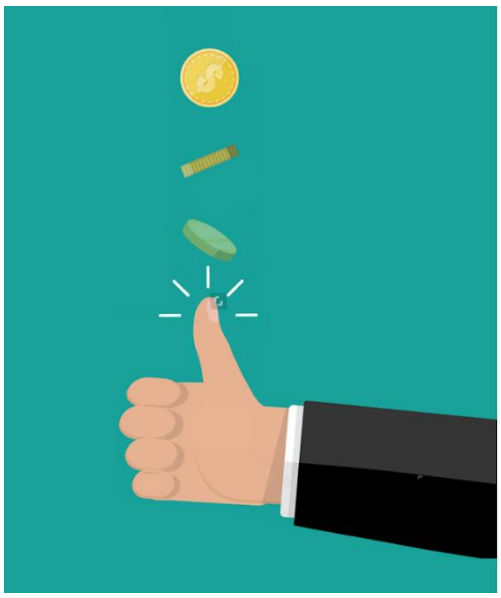
Determine whether the coin is fair or unfair based on the # of heads and # of tails when you toss the coin.

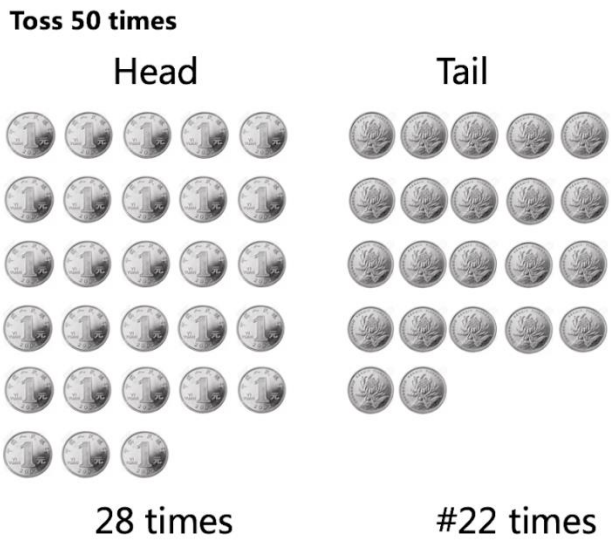
Okay, let me ask this question in another way. Let's say, if I give you a normal coin. Remember! It is a normal coil with one head and one tail. How many # of heads and # of tails will you get if you toss it 50 times?

Based on logic, the best case is 25 heads and 25 tails.

But I don't believe it will get the perfect scenario of 25 heads vs 25 tails.

Then, you will start to think, 28 heads and 22 tails are okay, and 23 heads and 27 tails are fine as well.





Here is the formula for the Chi-Squared test:

experimental #times

theoretical #times
if the coin is balanced

$$X^2 = \sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

theoretical #times
if the coin is balanced

1. The value of x^2

2. The Degrees of Freedom (if you don't know what is DoF, please check my DoF article in my blog)

3. Confidence Level (usually 90% or 95%)

投50次

Table (2rows*2columns)	Head	Tail
Expected times (E)	25	25
Observed times (O)	28	22

$$\frac{(O-E)^2}{E} + \frac{(O-E)^2}{E}$$

①chi-squared $\frac{(28-25)^2}{25} + \frac{(22-25)^2}{25} = \mathbf{0.72}$

②Degree of freedom (#row-1) * (#column-1) = (2-1) * (2-1) = **1**

③ significant level **95%**

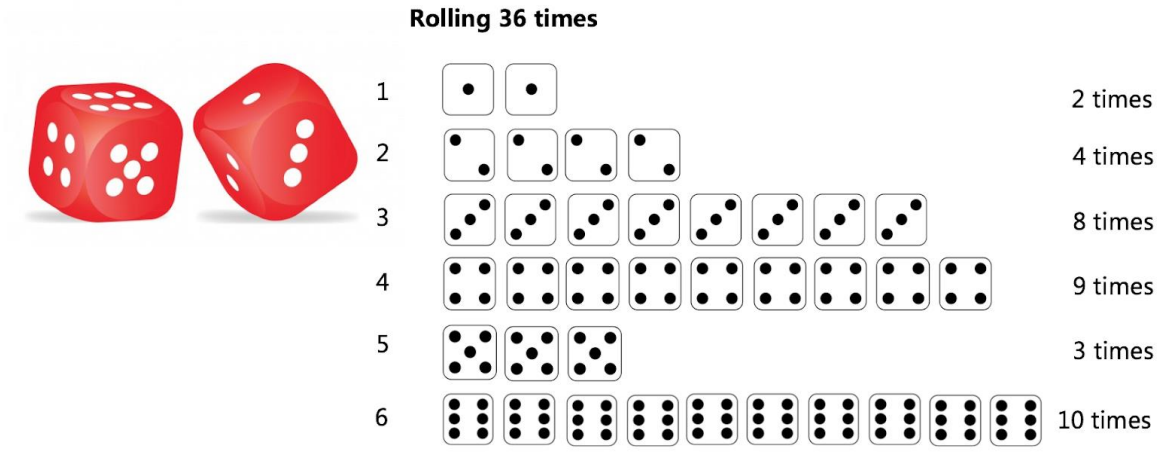
Once we get these three values then we could look up Chi-Squared table, because 0.72 is less than 3.84, we can not reject null hypothesis, so we can make the conclusion that the coin is fair.

	Probability level (alpha)					
df	0.5	0.10	③ 0.05	0.02	0.01	0.001
② 1	0.455	2.706	① 3.841	5.412	6.635	10.827
2	1.386	4.605	5.991	7.824	9.210	13.815
3	2.366	6.251	7.815	9.837	11.345	16.268
4	3.357	7.779	9.488	11.668	13.277	18.465
5	4.351	9.236	11.070	13.388	15.086	20.517

Conclusion: 0.72(chi-squared) < 3.841(threshold)
-> The coin is balanced (can't reject null hypothesis)

balanced !

Another example rolling dice:



Rolling 36 times

Table : 2rows*6 columns	1	2	3	4	5	6
Expected(E)	6	6	6	6	6	6
Observed(O)	2	4	8	9	3	10

$$\frac{(O-E)^2}{E} + \frac{(O-E)^2}{E} + \frac{(O-E)^2}{E} + \frac{(O-E)^2}{E} + \frac{(O-E)^2}{E} + \frac{(O-E)^2}{E}$$

$$\frac{(2-6)^2}{6} + \frac{(4-6)^2}{6} + \frac{(8-6)^2}{6} + \frac{(9-6)^2}{6} + \frac{(3-6)^2}{6} + \frac{(10-6)^2}{6}$$

①Chi-Squared = **9.6**

②Degree of freedom (#rows- 1) * (#columns– 1) = (2-1) * (6-1) = **5**

③ Significant level **90%**

	Probability level (alpha)					
df	0.5	0.10	0.05	0.02	0.01	0.001
1	0.455	2.706	3.841	5.412	6.635	10.827
2	1.386	4.605	5.991	7.824	9.210	13.815
3	2.366	6.251	7.815	9.837	11.345	16.268
4	3.357	7.779	9.488	11.668	13.277	18.465
5	4.351	9.236	11.070	13.388	15.086	20.517

Conclusion: 9.6(chi-squared) > 9.236(threshold)
-> The dice is unbalanced (can reject null hypothesis)

Unbalanced !

When you discuss a report with data analyst or data scientist next time, they tell you: hey, this variable is not significant, I removed from this analysis.

Then, you can ask him with confidence, what is the Chi-Squared value? What's your confidence level did you choose?

Gini is intended for continuous attributes and Entropy is for attributes that occur in classes

Gini is to minimize misclassification
Entropy is for exploratory analysis
Entropy is a little slower to compute

As per parsimony principal Gini outperform entropy as of computation ease (log is obvious has more computations involved rather that plain multiplication at processor/Machine level).

But entropy definitely has an edge in some data cases involving high imbalance.

Since entropy uses log of probabilities and multiplying with probabilities of event, what is happening at background is value of lower probabilities are getting scaled up.

If your data probability distribution is exponential or Laplace (like in case of deep learning where we need probability distribution at sharp point) entropy outperform Gini.

To give an example if you have 2 events one .01 probability and other .99 probability.

In Gini Prob sq will be .01^2+.99^2, .0001 + .9801 means lower probability do not play any role as everything is govern by majority probability.

Now in case of entropy .01*log(.01)+.99*log(.99)= .01*(-2)+ .99*(-.00436) = -.02-.00432
now in this case clearly seen lower probabilities are given better weight-age.

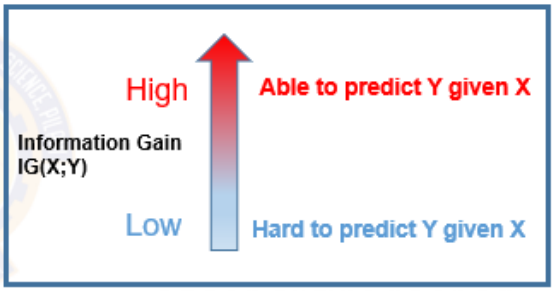
most information-theoretic concepts can only be applied to discrete variables. Therefore, feature selection algorithms in this family can only work with discrete data. For continuous feature values, some data discretization techniques are required beforehand.

MI is capable of measuring a general dependence between two features without assuming the distributions of the features

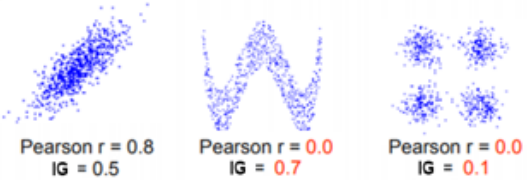
We want to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned. • Information gain tells us how important a given attribute of the feature vectors is.

Information Gain

Information Gain IG(X;Y) is a measure of the mutual independence between two random variables X and Y.



✓Symmetric I(X,Y) = I(Y,X)
✓Measures non-linear dependencies
✓I(X;Y)=0 if X and Y are independent
✓Biased towards the features having large number of values



Information Gain

Information Gain IG(X;Y) is a measure of the mutual independence between two random variables X and Y.

$$IG(X;Y) = H(Y) - H(Y|X)$$

$$= \sum_{x_i \in X} \sum_{y_j \in Y} \underbrace{P(x_i, y_j)}_{\text{joint probability mass function}} \log \frac{P(x_i, y_j)}{\underbrace{P(x_i)P(y_j)}_{\text{marginal probabilities}}}$$

Gain Ratio

➢ **The gain ratio "normalizes" the information gain**

$$Gain\ Ratio(Attribute) = \frac{IG(Attribute)}{H(Attribute)}$$

✓reduces the bias toward attributes with many values.

✓The feature with the maximum gain ratio is selected as the best feature.

The information gain ratio is a variant of the mutual information.

It can be seen as a normalization of the mutual information values from 0 to 1. It is the ratio of information to the entropy of the target attribute. By doing so, it also reduces the bias toward attributes with many values.

v Gain ratio: a modification of the information gain that reduces its bias v Gain ratio takes number and size of branches into account when choosing an attribute q It corrects the information gain by taking the intrinsic information of a split into account v Intrinsic information: entropy of distribution of instances into branches (i.e. how much info do we need to tell which branch an instance belongs to)

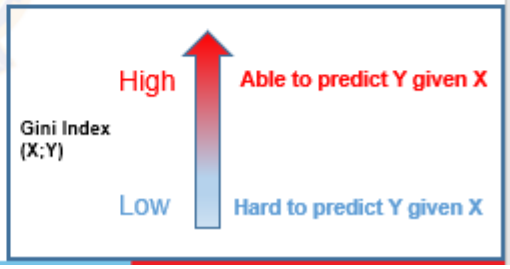
The information gain ratio is a variant of the mutual information. It can be seen as a normalization of the mutual information values from 0 to 1. It is the ratio of information to the entropy of the target attribute. By doing so, it also reduces the bias toward attributes with many values. There exists several versions of the definition of this measure. Especially, there exists an asymmetric and a symmetric version.

Gini Index

➢Gini index minimize the probability of misclassification

$$\text{Gini} = 1 - \sum_{i=1}^{K} {p_k}^2$$

where p_k denotes the proportion of instances belonging to class k K = 1, … , k.



Example

ATTRIBUTES				CLASS LABEL
Gender	**CAR Ownership**	**Travel Cost(Rs/Km)**	**Income Level**	**Transport Mode**
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car
Female	1	Cheap	Medium	Train
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train

Transport Mode		
Bus	**Train**	**Car**
4	3	3

H(Transport Mode) = H(4,3,3)
$= -(4/10 \log_2 4/10) - (3/10 \log_2 3/10) - (3/10 \log_2 3/10)$
= 1.571

\

Example (Contd..)

		Class Label		
		Bus	Train	Car
Attribute values	Cheap	4	1	0
	Expensive	0	0	3
	Standard	0	2	0

H(Transport Mode)= 1.571

IG(Transport Mode, Travel Cost) = H(Transport Mode)-H(Transport Mode| Travel Cost)
= 1.571 - H(4,3,2)
= 1.571 - (-5/10 (4/5 log 4/5 +1/5log1/5) - (3/10 (3/3 log 3/3 + 0 log 0)) - (3/10 (2/2 log 2/2 + 0 log 0)))
= 1.571 – 0.36 = 1.211

Gain Ratio (Travel Cost) = $\frac{IG(Attribute)}{H(Attribute)} = \frac{1.211}{H(Attribute)}$
H(Attribute) = -(5/10 log 5/10)-(3/10 log 3/10)- (2/10 log 2/10) = 1.48
$= \frac{1.211}{1.48} = 0.818$

Gini Index(Transport Mode,|Travel Cost=cheap) = 1-(0.8^2+0.2^2) = 0.32
Gini Index(Transport Mode,|Travel Cost=Expensive) = 1-(1^2+0^2)=0
Gini Index(Transport Mode,|Travel Cost=Standard) = 1-(1^2+0^2)=0
Gini Index(Transport Mode,|Travel Cost=cheap) = 5/10*0.32+3/10*0+2/10*0 = 0.16

Example (Contd..)

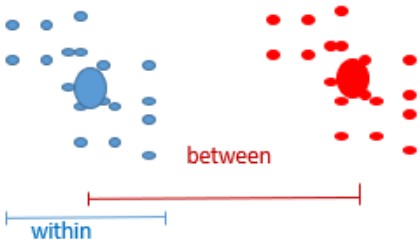
	Information Gain	Gain Ratio	Gini
Gender	0.147 4	0.147 4	0.6 1
Car Ownership	0.544 3	0.368 3	0.453 2
Travel Cost	1.21 ✓ 1	0.818 ✓ 1	0.16 4
Income Level	0.696 2	0.458 2	0.366 3

Fischer score:

- Define the inter and intraclass distances
- Formulate the Fischer score filter

How class information is useful?

- Applicable for classification problems with numeric features.
- Between-class distance – Distance between the centroids of different classes
- Within-class distance – Accumulated distance of an instance to the centroid of its class



Features or the outcome variable must be discretized before applying filter to real-valued features in a prediction (regression) problem

Metrics can be applied naturally to real-valued features in a binary classification problem

Metrics can be applied naturally to real-valued features in a multi-class classification problem

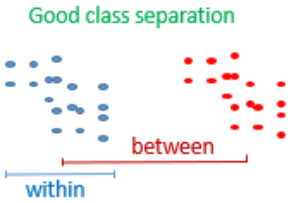
Fisher Score

- Fisher score is the measure the ratio of the average interclass separation to the average intraclass separation.
- The larger the Fisher score, the greater the discriminatory power of the attribute.

$$F = \frac{\sum_{j=1}^{k} p_j(\mu_j - \mu)^2}{\sum_{j=1}^{k} p_j \sigma_j^2}$$

μ_j - mean of the data points belonging to class j for a particular feature,
σ_j - standard deviation of data points belonging to class j for a particular feature,
p_j - the fraction of data points belonging to class j.
μ - the global mean of the data on the feature

- This score is often referred as signal to noise ratio



Fisher's ratio is a measure for (linear) discriminating power of a variable

- Maximum between class variance (difference of means)
- Minimum within class variance (sum of variances).

The Fisher Ratio is defined as the ratio of the variance of the between classes to the variance of within classes

Because the score of each feature is computed independently, the features selected by the heuristic algorithm is suboptimal. More importantly, as we mentioned before, the heuristic algorithm fails to select those features which have relatively low individual scores but a very high score when they are combined together as a whole. In addition, it cannot handle redundant features.

Wrapper Based Methods

- Greedy Based algorithms
- Agnostic to the machine learning models chosen.
- Sequential feature selection algorithm add or remove one feature at a time based on the classifier performance until a desired criterion is met.

Algorithm for Wrapper based method

Input: large feature set Ω

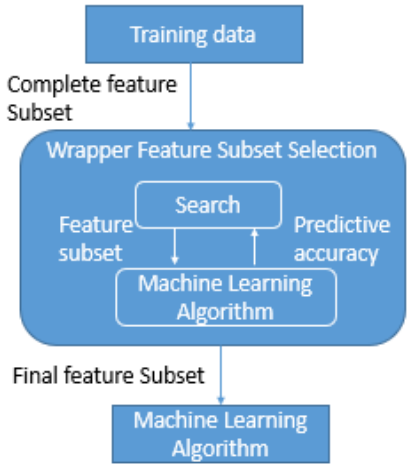
Identify candidate subset S ⊆ Ω

While !stop_criterion()

Evaluate error of a classifier using S.

Adapt subset S.

Return S.

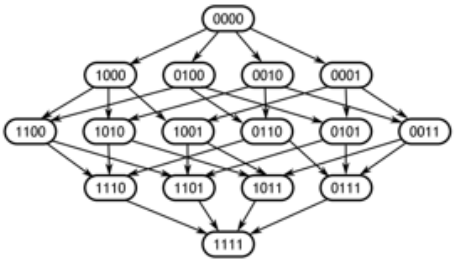


- Commonly used Stop criterions
  - ✓Increase / Decrease in Predictive accuracy
  - ✓Predefined number of features is reached

WRAPPER BASED FEATURE SUBSET SELECTION: In this methods the search algorithm outputs a feature subset which is again used by the ML algorithm and the prediction accuracy is obtained which is fed to the search algorithm and then it proceeds and finally it outputs the final subset.

Sequential forward selection(SFS)

1. Start with the empty set $Y_0 = \{\emptyset\}$
2. Select the next best feature $x^+ = \underset{x \notin Y_k}{\arg\max}\, J(Y_k + x)$
3. Update $Y_{k+1} = Y_k + x^+$; $k = k + 1$
4. Go to 2



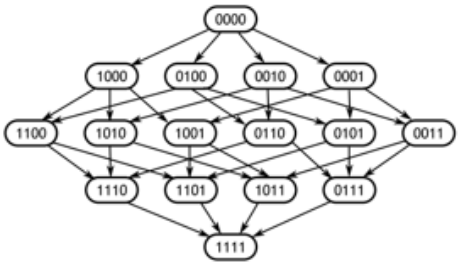
Step for

If features are redundant you may not use all of them hence we can find all uncorrelated features. You can arbitrarily start with a feature and keep adding other feature and if you find correlation with the new added feature we can discard them.

We also have to eliminate irreverent features

Sequential Backward selection(SBS)

1. Start with the full set $Y_0 = \{X\}$
2. Remove the worst feature $x^- = \underset{x \in Y_k}{\arg\max}\, J(Y_k - x)$
3. Update $Y_{k+1} = Y_k - x^-$; $k = k + 1$
4. Go to 2



- Backwards selection is frequently used with random forest models

recursive feature elimination

Therefore, the subset size is a *tuning parameter* for RFE.

Pros and Cons of Greedy Sequential Algorithms

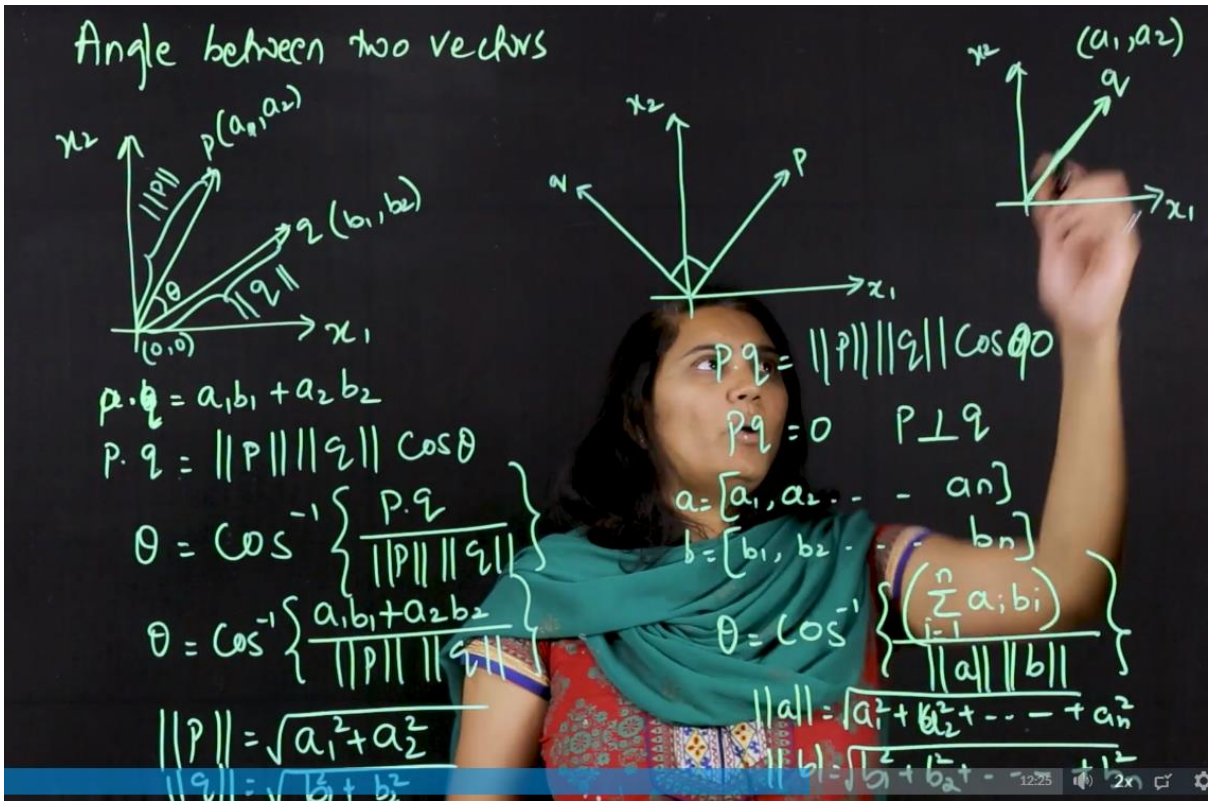
- Pros
  - ✓ Highest performance
- Cons
  - ✓ Computationally expensive
  - ✓ Memory intensive

Module 3:

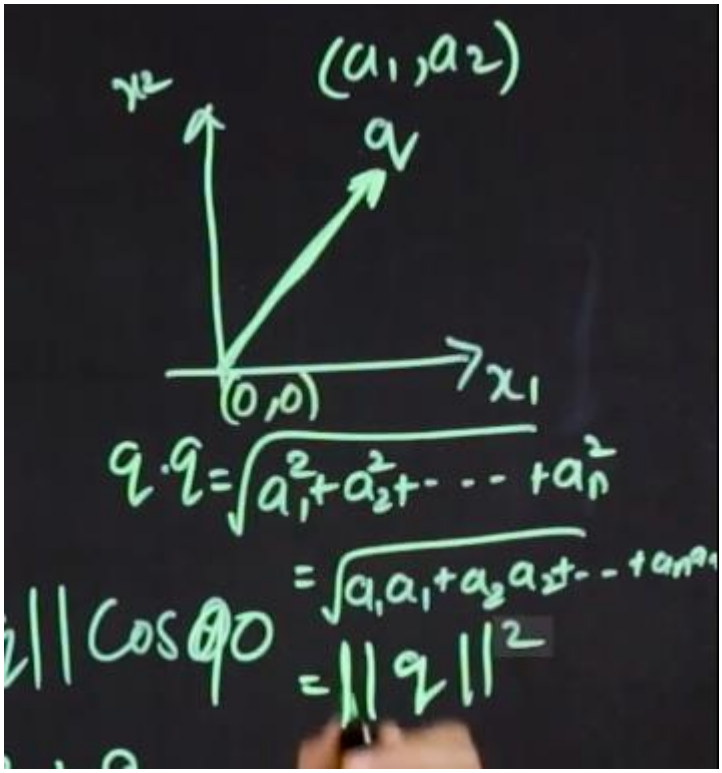
Statistics:

Dot product of 2 vectors is denoted as a.b which is basically a a^T x B. The result is scalar.

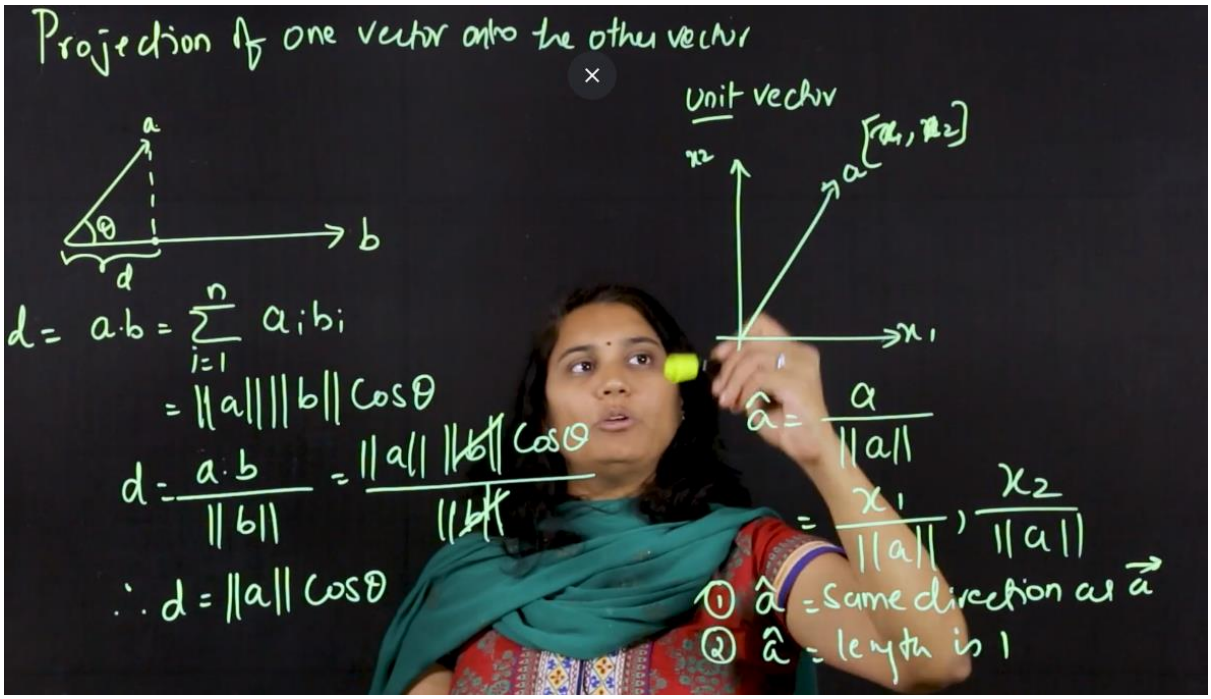
The Cross Product **a × b** of two vectors is **another vector** that is at right angles to both

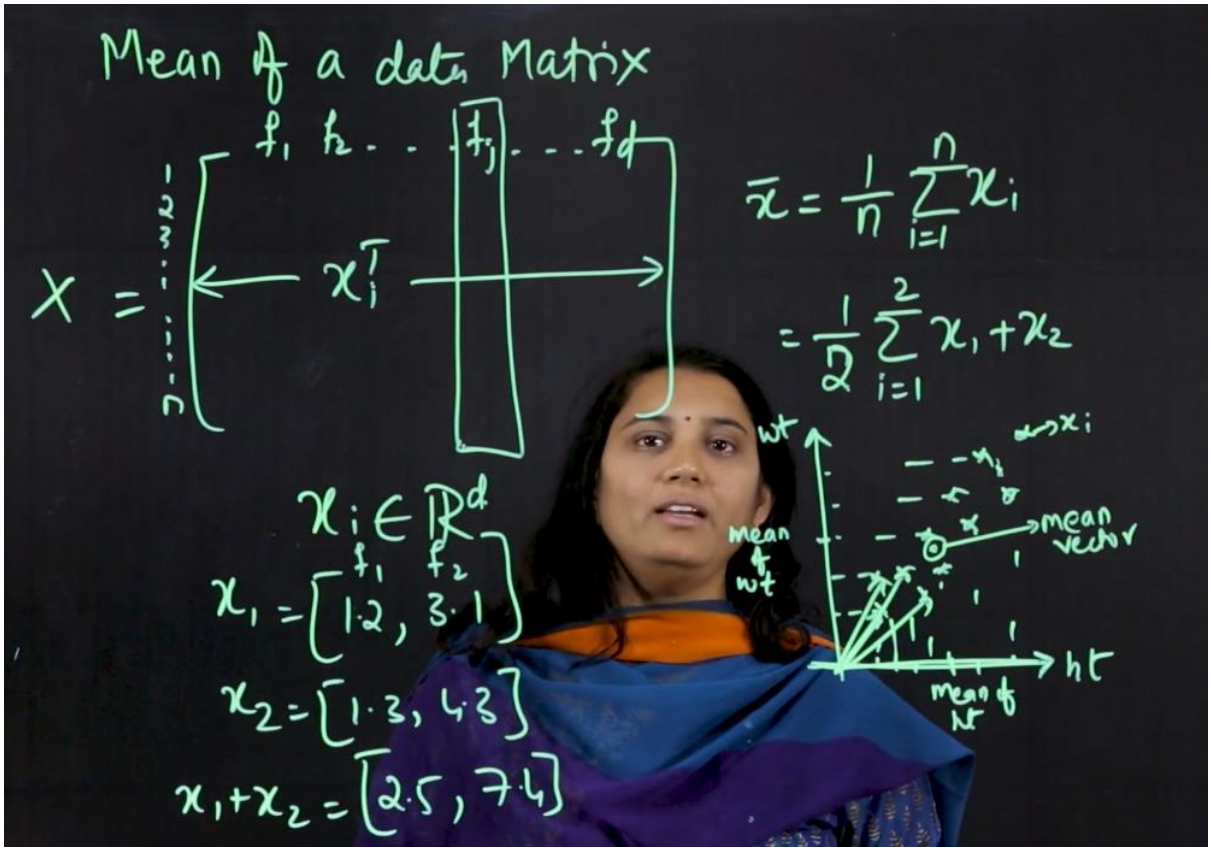


If dot product is zero, then we can conclude that they are perpendicular.



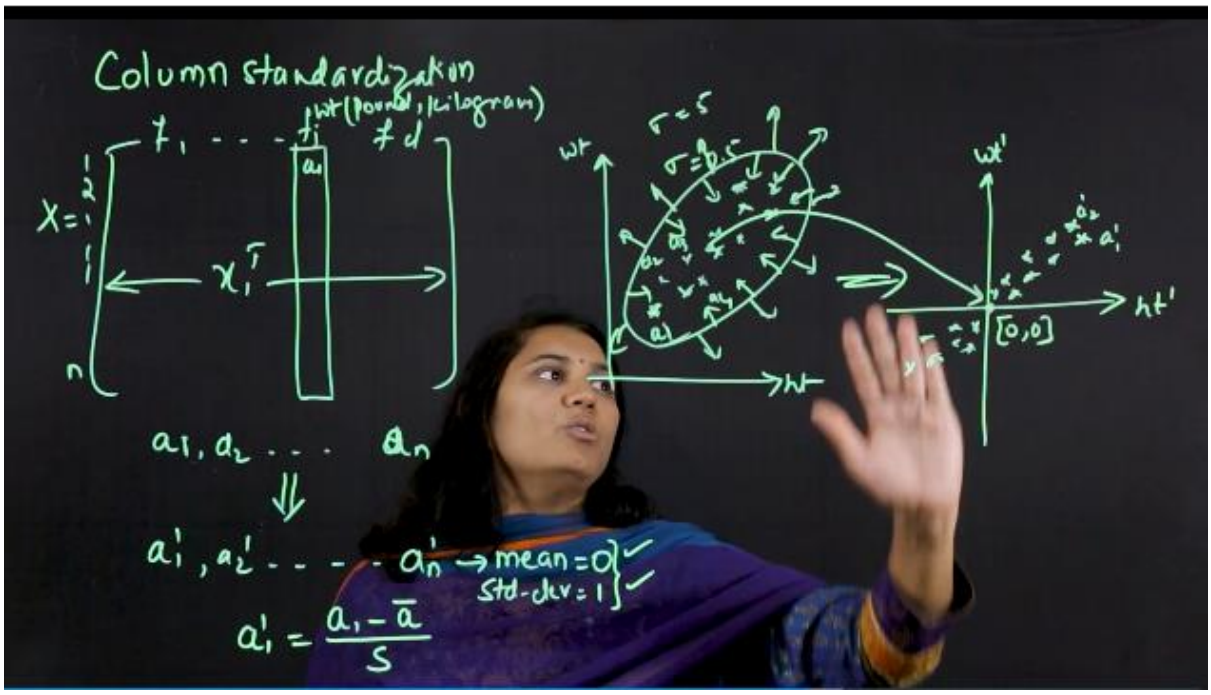
Angle between a vector to itself is its dot product as shown above.

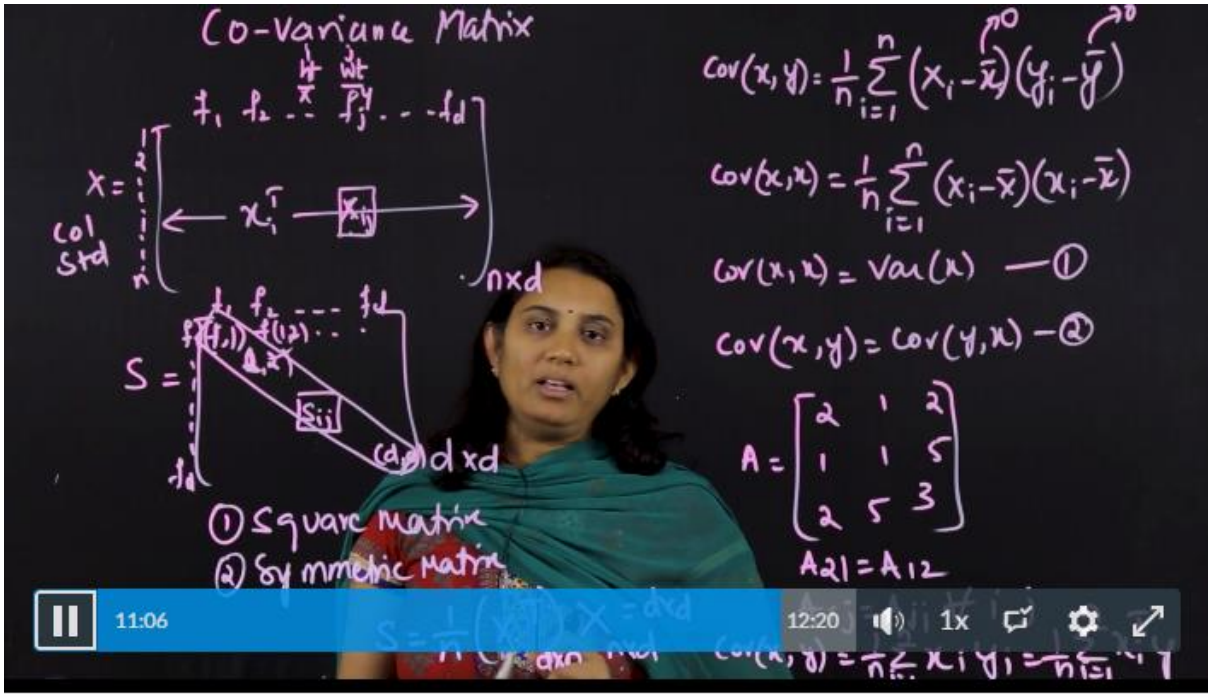




Mean vector is used extensively.

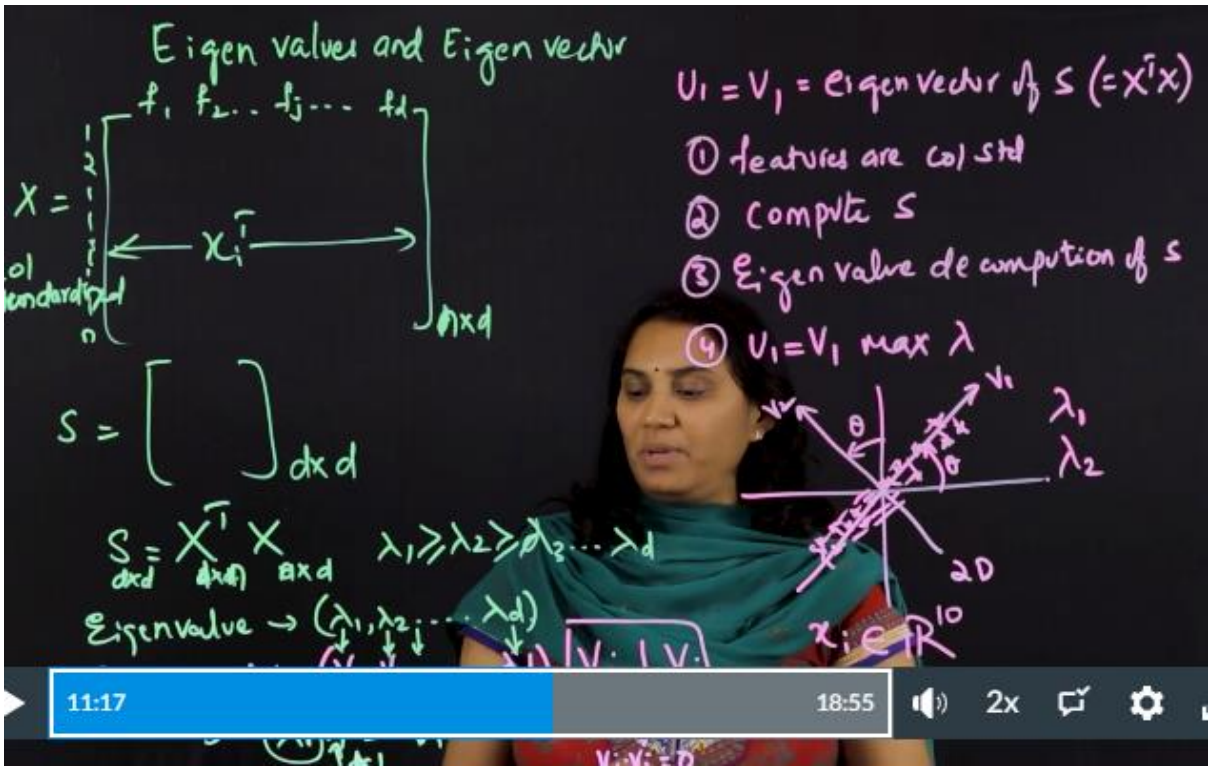
Column standardization

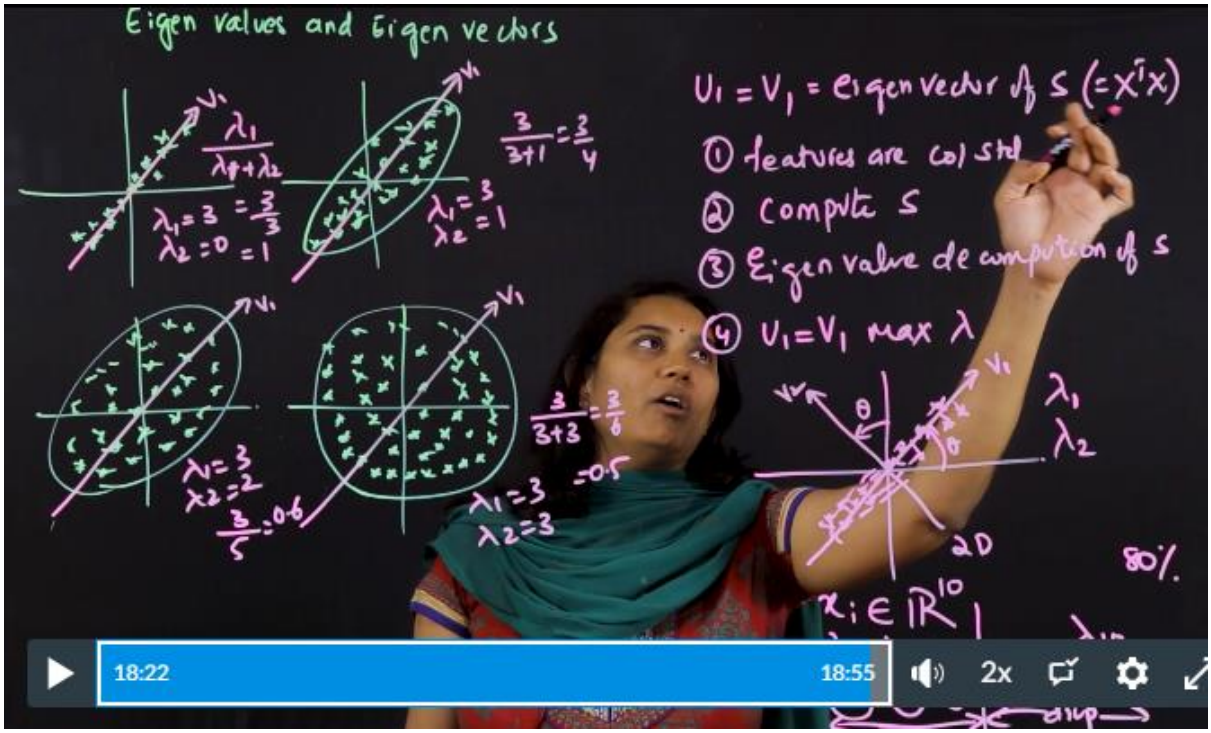




For column standardized vectors, since mean of x is zero, the above equation has changed.

In simple words, both the terms measure the relationship and the dependency between two variables. “**Covariance**” indicates the direction of the linear relationship between variables. “**Correlation**” on the other hand measures both the strength and direction of the linear relationship between two variables.





Module 3:

Combinations: Order does not matter. Choose 3 apples from 10

$$C(n,r) = \frac{n!}{(r!(n-r)!)}$$

Permutations: Order does matter(like choose president, ceo, manager from group)

$$P(n,k) = \frac{n!}{(n-k)!}$$

Combinations are always fewer than permutations.

Scatter plot (2 variables) and pair plots (more than 2) are used to depict the relationships between variables.

nC2 is the no. of scatter/pair plots we need to draw.

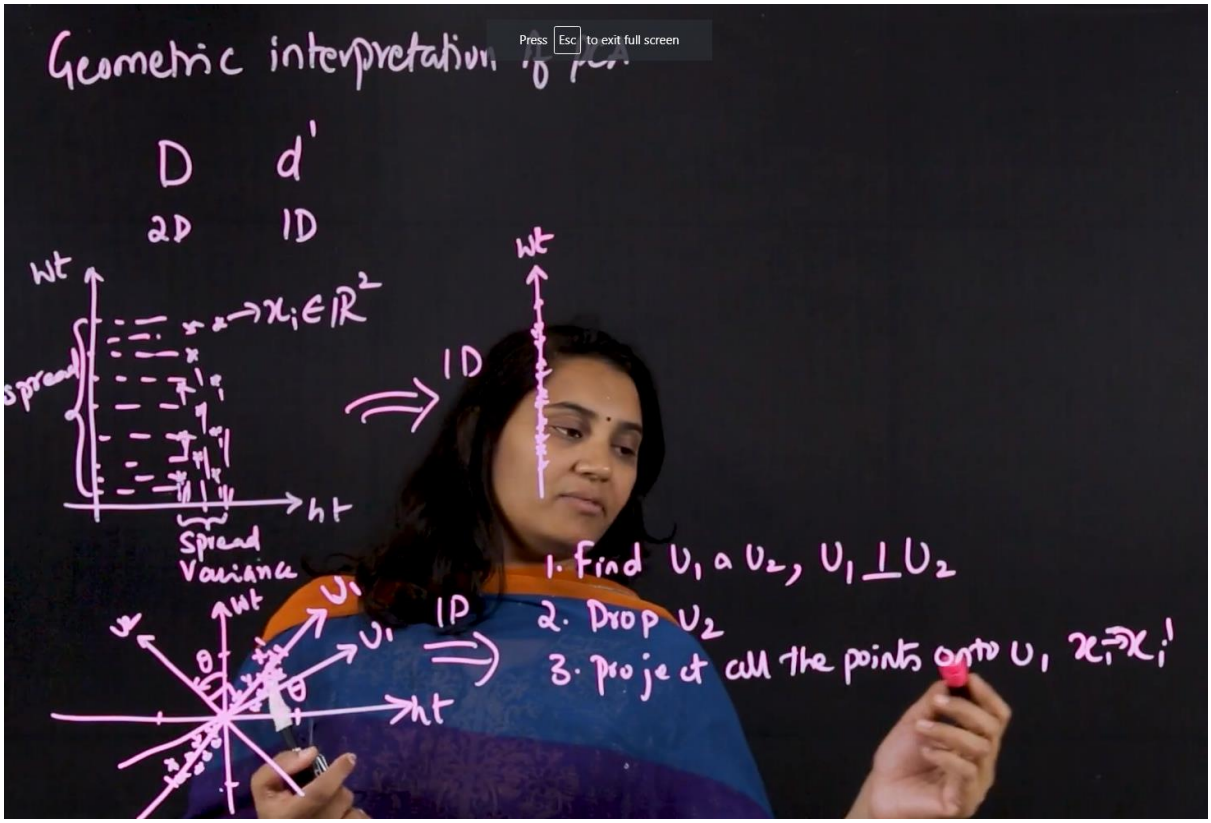
We want to reduce the dimensions from higher dimensions to lower dimensions to represent the data easily.

In case of weight and height, we can represent both of them using BMI.

Similarly, highly correlated features can be removed from the feature set.

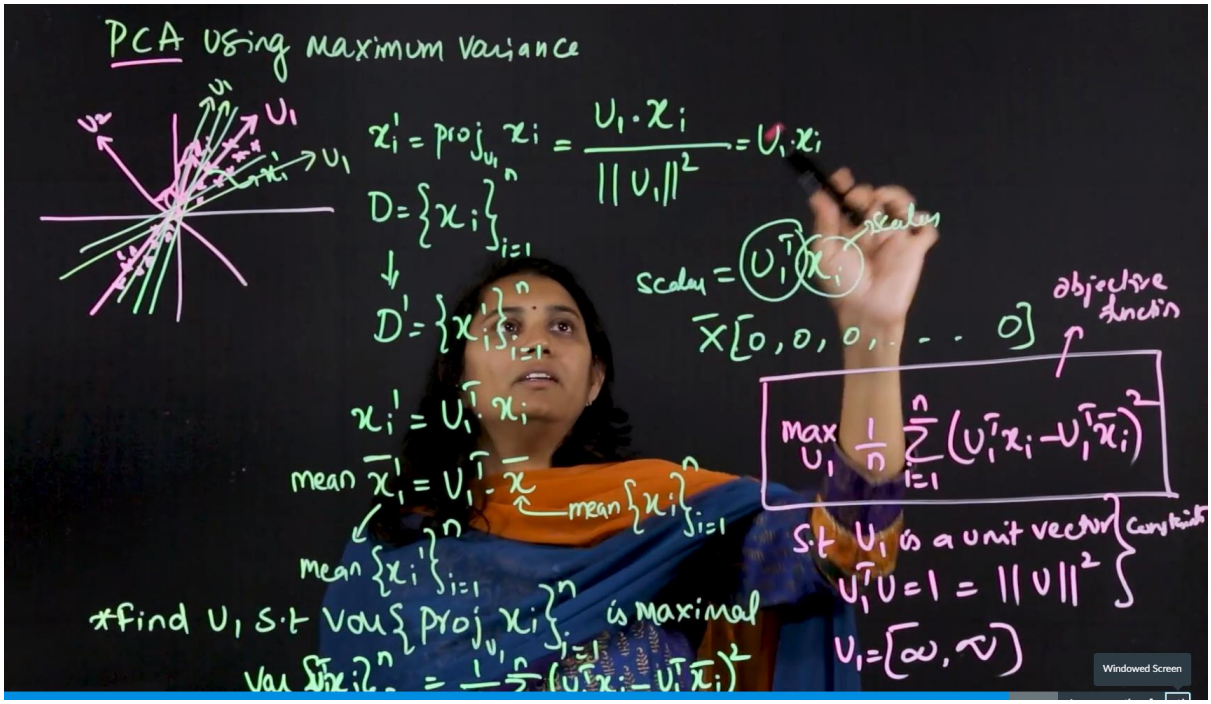
We use two techniques for dimensionality reduction.

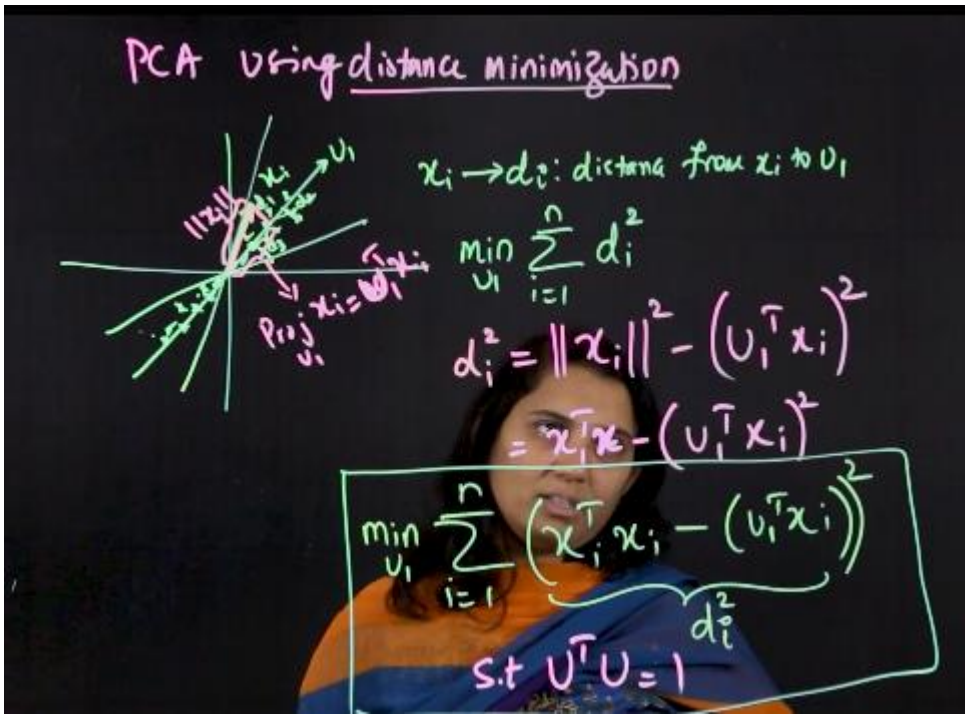
1. PCA
2. Feature Selection



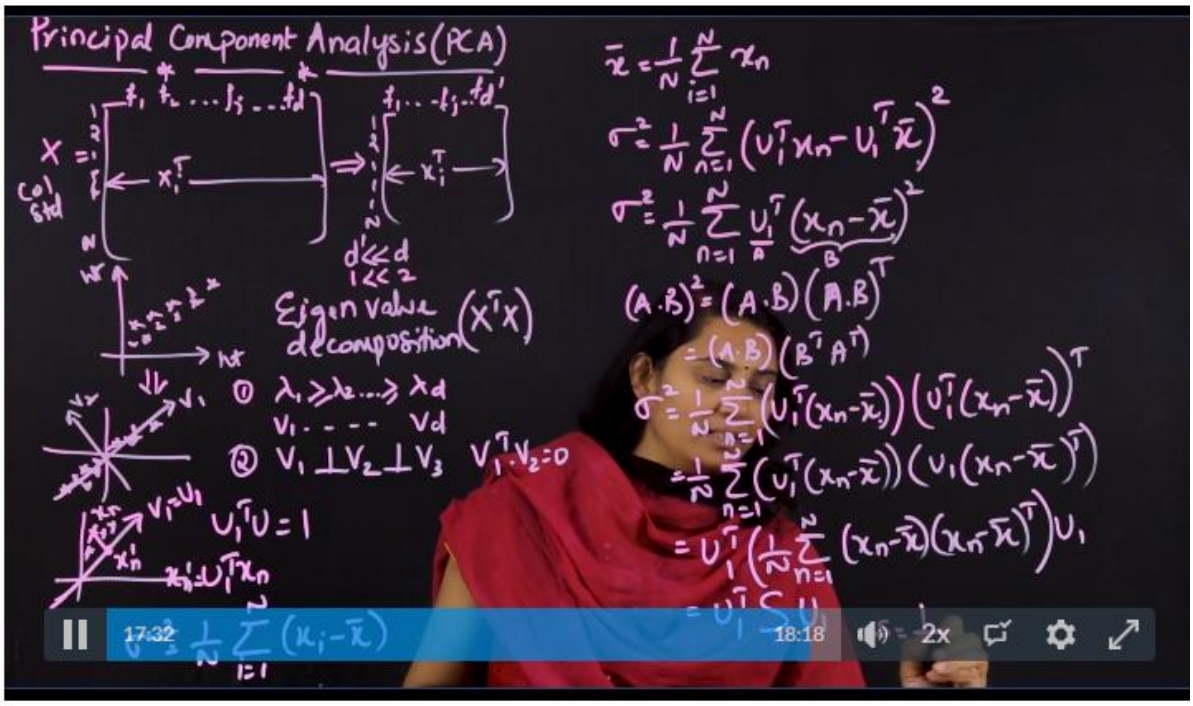
Here for height and weight, the spread/variance is more on t axis, so, we will keep weight and remove height.

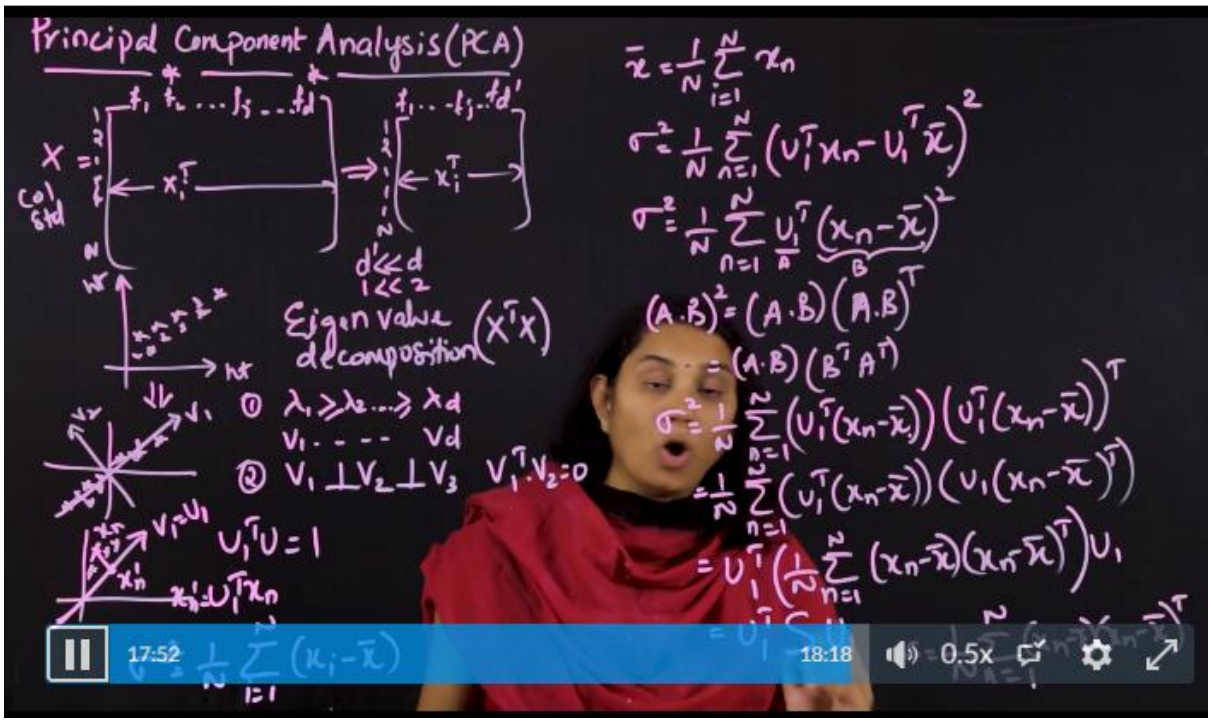
Similar thing we need to do for any features. Here we will find U1 vector that maximizes variance and find another vector which is perpendicular to this vector and then drop it. Then project all point onto U1.

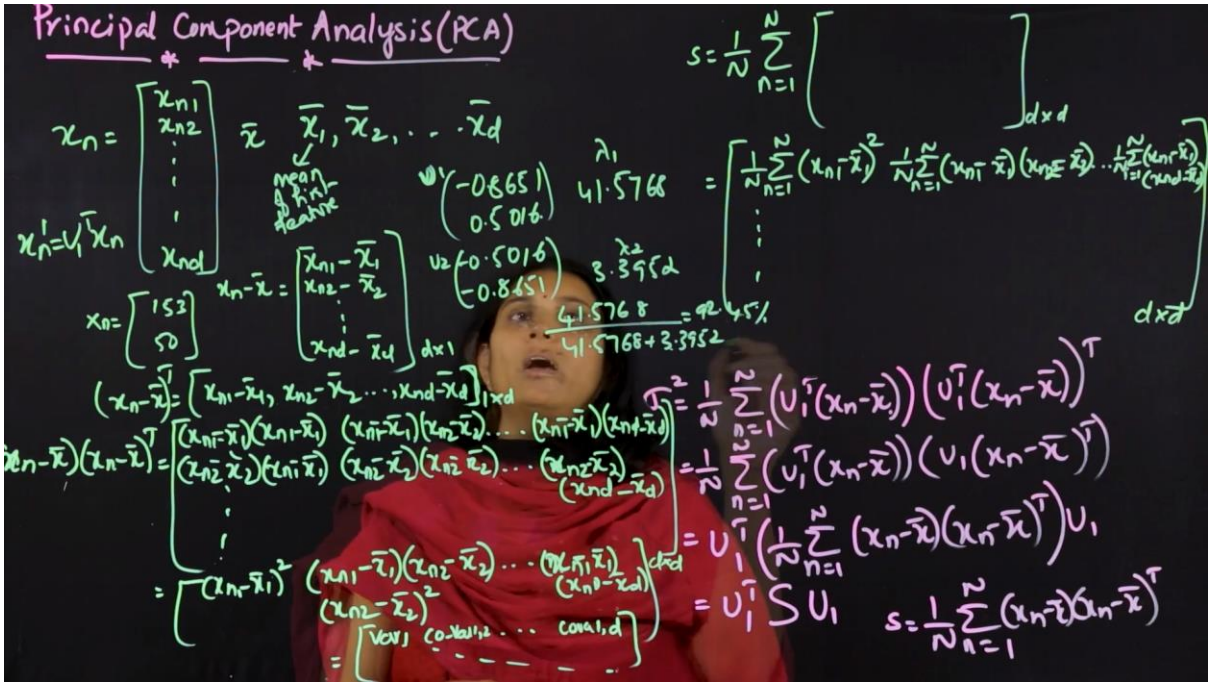




PCA could be done using either maximum variance or minimum distance.


Principal Component Analysis (PCA)
Eigen value decomposition $(X^T X)$
① $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_d$
$v_1 \ldots v_d$
② $v_1 \perp v_2 \perp v_3$ $v_1^T v_2 = 0$
$v_1 = u_1$ $u_1^T u = 1$
$x_n' = u_1^T x_n$
$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_n$
$\sigma^2 = \frac{1}{N} \sum_{n=1}^{N} (u_1^T x_n - u_1^T \bar{x})^2$
$\sigma^2 = \frac{1}{N} \sum_{n=1}^{N} u_1^T (x_n - \bar{x})^2$
$(A.B)^2 = (A.B)(A.B)^T$
$= (A.B)(B^T A^T)$
$\sigma^2 = \frac{1}{N} \sum_{n=1}^{N} (u_1^T (x_n - \bar{x}))(u_1^T (x_n - \bar{x}))^T$
$= \frac{1}{N} \sum_{n=1}^{N} (u_1^T (x_n - \bar{x}))(u_1 (x_n - \bar{x})^T)$
$= u_1^T \left(\frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T \right) u_1$
17:32
18:18
2x


Principal Component Analysis (PCA)
Eigen value decomposition $(X^T X)$
① $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_d$
$v_1 \ldots v_d$
② $v_1 \perp v_2 \perp v_3$ $v_1^T v_2 = 0$
$v_1 = u_1$ $u_1^T u = 1$
$x_n' = u_1^T x_n$
$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_n$
$\sigma^2 = \frac{1}{N} \sum_{n=1}^{N} (u_1^T x_n - u_1^T \bar{x})^2$
$\sigma^2 = \frac{1}{N} \sum_{n=1}^{N} u_1^T (x_n - \bar{x})^2$
$(A.B)^2 = (A.B)(A.B)^T$
$= (A.B)(B^T A^T)$
$\sigma^2 = \frac{1}{N} \sum_{n=1}^{N} (u_1^T (x_n - \bar{x}))(u_1^T (x_n - \bar{x}))^T$
$= \frac{1}{N} \sum_{n=1}^{N} (u_1^T (x_n - \bar{x}))(u_1 (x_n - \bar{x})^T)$
$= u_1^T \left(\frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T \right) u_1$
17:52
18:18
0.5x



Visualization:

Data Structure

Cross Sectional Data

- *Cross Sectional Data* is data collected at a **SINGLE** instance. This can be a transaction done at 4.00PM by 20000 employees along with information about their demographics, salary etc. Generally we perform *Prediction i.e. Machine Learning on* this kind of data

→ Predictions
Cause > Effect

Time Series Data

- *Time Series* is data collected at **MULTIPLE** instances (preferably periodic/cyclic). This can be transaction done at various time periods by a single employee WITHOUT any information about demographics or salary etc. Generally we perform *Forecasting* on this kind of data

→ Forecasting

Data Types

Generally,

Dimensions

- Dimensions are largely the categorical fields in the dataset. These variables helps to perform slice and dice on the dataset. In predictive analytics space, we execute classification algorithms on dependent dimensions (For Eg: Colors, Gender, Country, Risk Level, Priority etc.)

→ X-Axis

Measures

- Measures are largely the numerical fields in the dataset. These variables quantify each record thereby helps to compare multiple records quantitatively. In predictive analytics space, we perform descriptive statistical analysis (mean, sd, distribution etc.) and linear regression analysis on these variables (For Eg: Temperature, Salary, Transaction Amount etc.)

→ Y-Axis

Population Parameters & Sample Statistics

Descriptive Statistics

- Describe summary of data for better representation and understand patterns (if any)

Inferential Statistics

- Draw conclusions or predictions or generalizations about population based on the sample

Parameter

- Any statistical measure about population is termed as 'Parameter'

Statistic

- Any statistical measure about sample is termed as 'Statistic'

Analysis

- Approach to understand patterns in data, arrive at conclusions, evaluate hypothesis and build cause and effect relationships
- Cleaning > Summarization > Visualization
- Inference

Population Parameters & Sample Statistics

Univariate Analysis
- Describes one variable (Statistical Analysis, Visual Analysis)

Bivariate / Multivariate Analysis
- Empirical Relationship between two more more variables (Statistical / Visual)

Summarization
- Centrality
- Dispersion
- Concentration

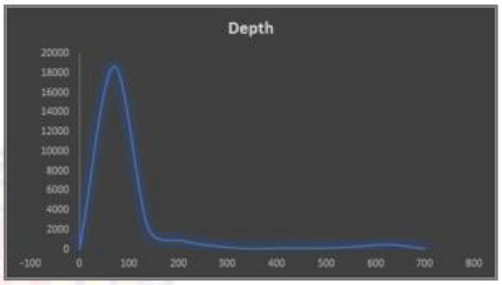
Centrality (Central Tendency)
- Summarize large data with central value
- Arithmetic Mean or Average or Mean
- Median
- Mode
- Quartiles

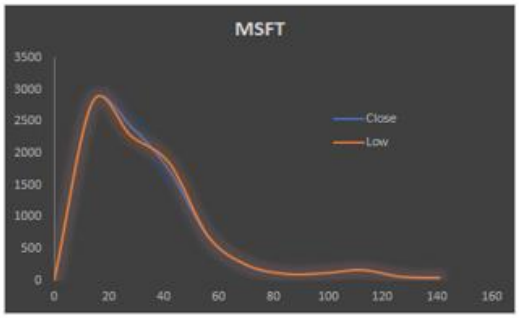
Dispersion
- Describes the spread of the data despite having similar central value
- Range
- Inter Quartile Range
- Variance
- Standard Deviation

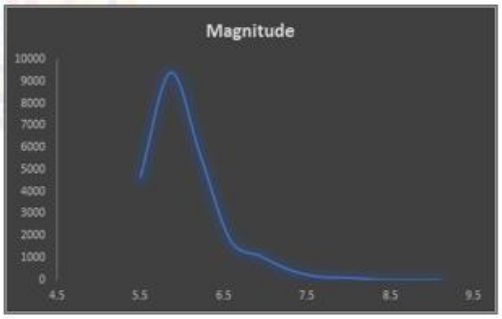
Concentration
- Describes where data density is high/low
- Higher Moments
- Skewness
- Kurtosis

Population Parameters & Sample Statistics – The Intuition & Why/How

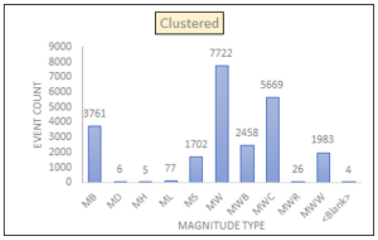
Term	Depth	Magnitude	
Count	23412	23412	
Average	70.76791	5.882531	
Std Dev	122.6493	0.423057	
Max	700	9.1	
Min	-1.1	5.5	
Range	701.1	3.6	
Variance	15042.85	0.178977	
Skewness	3.290472	1.848227	Measure Symmetricity
Kurtosis	10.45685	4.388922	Measure Peakedness
Q0	-1.1	5.5	Same as Minimum
Q1	14.5225	5.6	Value below which 1 Quarter of Datapoints lie
Q2	33	5.7	
Q3	54	6	
Q4	700	9.1	Same as Maximum
Correlation	0.023457312		

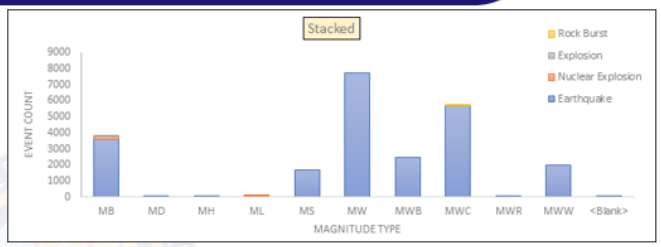


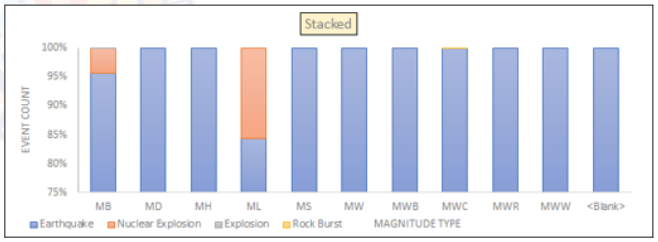




Column Chart and Bar Chart (Inverted Column Chart)







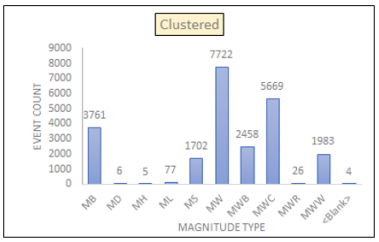
Approach and Benefits

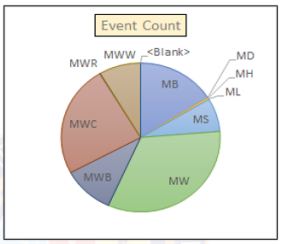
- Univariate Analysis
- Slice / Filter the data into 1 Dimension [Clustered]
- Slice / Filter the data into 2 Dimensions [Stacked]
- Slice / Filter the data into n Dimensions then use filters
- Used to identify which part of the LIST is highly dense (Popular : Pareto Analysis)
- Helps to build PREDICTIVE (Cause & Effect) Models

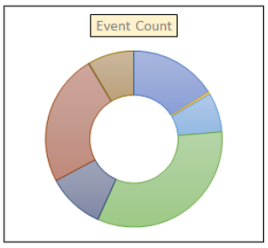
In box plot, the smaller the length of the box is, the lesser the spread is.. means the data is consistent.

Except scatter plots and contour plots, everything else is univariate.

Column Chart Vs Pie Chart Vs Heatmap / Tree map

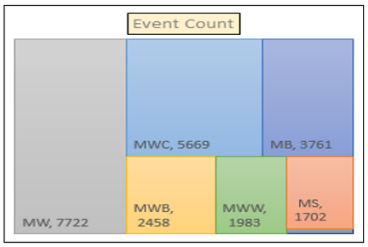




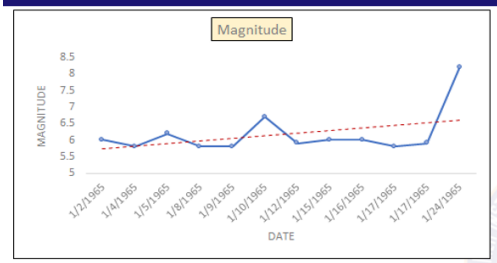


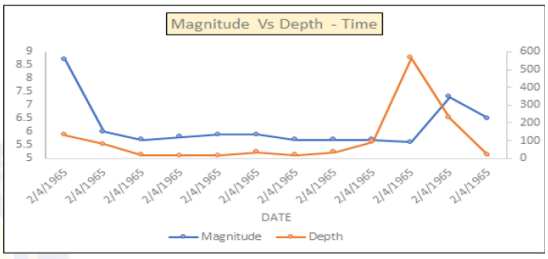
Approach and Benefits

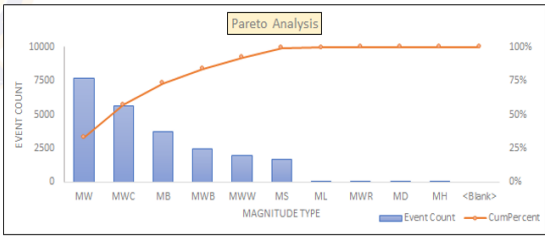
- Univariate Analysis
- Slice / Filter the data into 1 Dimension [Clustered]
- Can relatively understand which value of dimension contribute more, but VISUALLY tough for interpretation
- An advanced version of Pie Chart is Dognut chart & Treemap / Heatmap



Line Chart







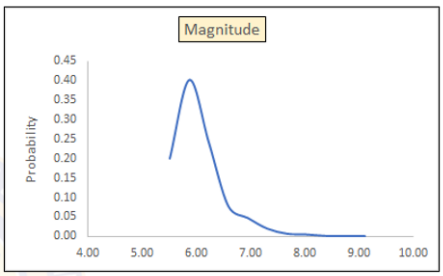
Approach and Benefits

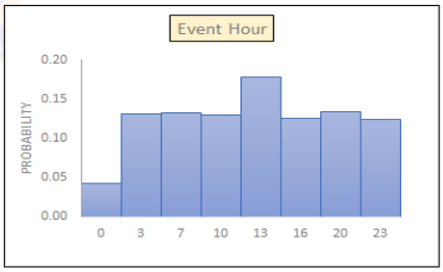
- Univariate Analysis
- X-Axis is LINE and Y-Axis is Measure
- We can show additional measures on secondary Y-Axis (Line or Bar), but in different color to have measures comparison [ON TIME LINE]
- We can fit best line (Regression) which provides FORECASTING MODEL
- We can show additional measures on secondary Y-Axis (Line or Bar), but in different color to have measures comparison [ON DIMENSIONS]

Histogram – Frequency / Probability Distribution

Approach and Benefits

- Univariate Analysis where X-Axis is the MEASURE, Y-Axis is Frequency of occurrence or probability
- Probability = Number of Favorable Events / Total Events
- Histograms visually provides various statistical measures BEYOND average
- Key Steps
  - Identify 'Variable of Interest' and evaluate Discrete / Continuous
  - Make x-axis as this measure
  - Decide the bins and compute frequency in each
  - Plot Bar for Discrete and Line for continuous

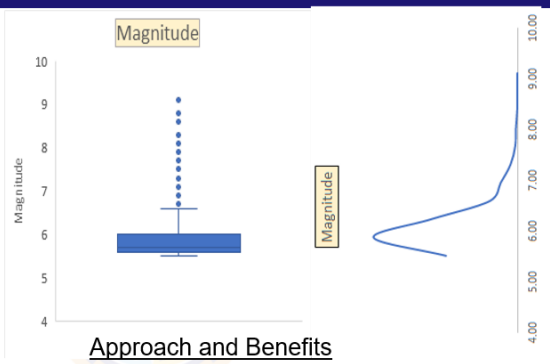




Box Plot - The Why and How

Approach and Benefits

- Summarize large amount of data
- Box Plot is viewing Histogram / Probability Distribution from 90°
- We get outliers from Box plot which should be analyzed and eliminated before USING for any modeling exercise
- MEDIAN is the statistical measure on which Box plot would be built
- We would come to know the symmetricity of the distribution using Box Plot
- X-Axis doesn't have any significance, but Y-axis is the range of measure
- Shorter line means <Dense>, Longer line means <Spread>

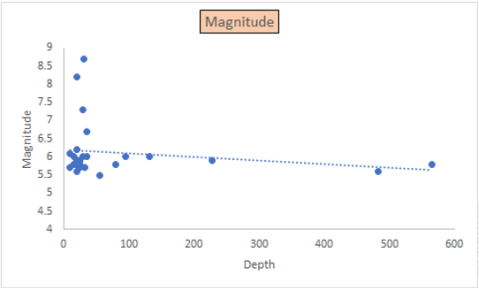


Approach and Benefits

Steps to build BOX Plot

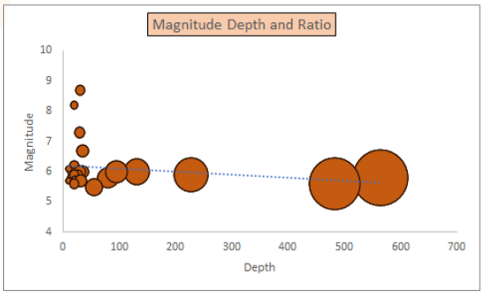
- Sort the data in ascending order
- Compute *Median (C)*
- Compute *Median* of *Lower (B) Spilt* & *Upper (D) Spilt* of entire data
- Compute *Min (A)* and *Max (F)* of entire Data
- Finally you have A, B, C, D, E. Plot these on number line and create BOX between BCD
- Quartiles *[Q1 < B, C > Q2 > B, D > Q3 > C, D > Q4]*

Scatter and Bubble Chart - The Why and How

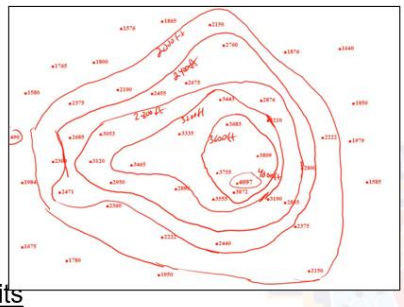


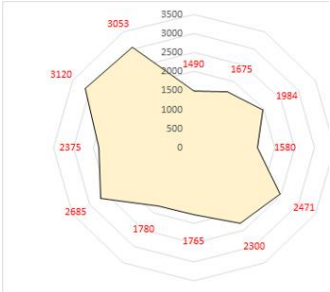
Approach and Benefits

- Bivariate Analysis where X-Axis is measure1 and Y-Axis is measure2
- Statistical measure for scatter plot is Correlation Coefficient
- Helps to build prediction models
- Helps to reduce model complexities by eliminating dependent variables (independent)
- Can be extended with additional measure3 for size of dots



Contour Plot - The Why and How





Approach and Benefits

- Contour plots are helpful when we have x and y as spatial coordinates and z as a measure that varies across these spatial coordinates
- It is the next version to scatter plot, where in each dot is labelled with z value
- Then, the z values are grouped (like Bins) and each group (say z-High, z-Low) are bounded such that INSIDE values satisfy the group condition

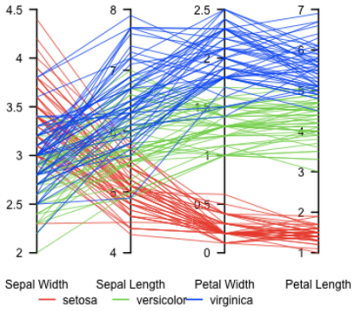
Steps To Follow

- Scatter plot on X- and Y- data
- Label Z for each dot
- Bucket the Z values into 5 to 7 groups and draw boundaries

Note: We have a minor version of contour map in excel - Radar

For multi-variate plotting, use parallel coordinates. This is a part of pandas plotting library.

Parallel Coordinates - The Why and How



Approach and Benefits

- Helps to plot high dimensional data in a single plot
- Each vertical lines represents a measure (E.g. Salary, Experience)
- Each line represents a record (E.g. Emp1, Emp2, and so on)
- Each color represents a class (Male, Female)

Difference from Line Chart

- Parallel Coordinates is similar to line plot, but with additional y-axes's

Note

- Recommended to scale the measures to z-score before plotting parallel coordinates. This is to ensure scaling challenge between measures is normalized

Steps to Follow

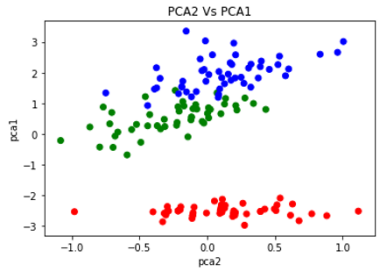
- Let's say you have 3 measure (Salary, Experience and Relevant Experience)
- Draw Y=1, Y=3, Y=5 as 3 horizontal lines for each of above measures
- Normalize all above measures to z-score (-1 to 1)
- Plot all 3 measure values on Y=1, Y=3 and Y=5 lines
- Connect these dots on 3 horizontal lines for EACH record
- Color the lines based on GENDER (Classification or Dimension)

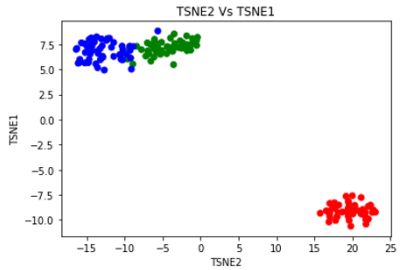
t-SNE - The Why and How

Problem
- Big Data is problem
- Having many details is the problem
- Incapability to visualize beyond 2 dimensions is the problem
- Redundancy in features is the problem
- Unable to justify if some feature must be removed or retained

Dimensionality Reduction
- Retain all features
- Reduce variables or features statistically or probabilistically
- PCA : Principal Component Analysis
- T-SNE : T Distribution Stochastic Neighbourhood Embedding

Points To Note
- Non linear relationships
- This is based in probability (Similarity), where as PCA is based on statistics (Covariance Maximization)
- T-SNE - Data > Pairwise Similarity between data points > Dimensional Reduction (Rather Selecting 2)
- PCA – Data > Covariance between features > Dimensional Reduction





PCA is a statistical technique that considers the features

t-SNE is a probabilistic technique that considers the observations.

PCA:

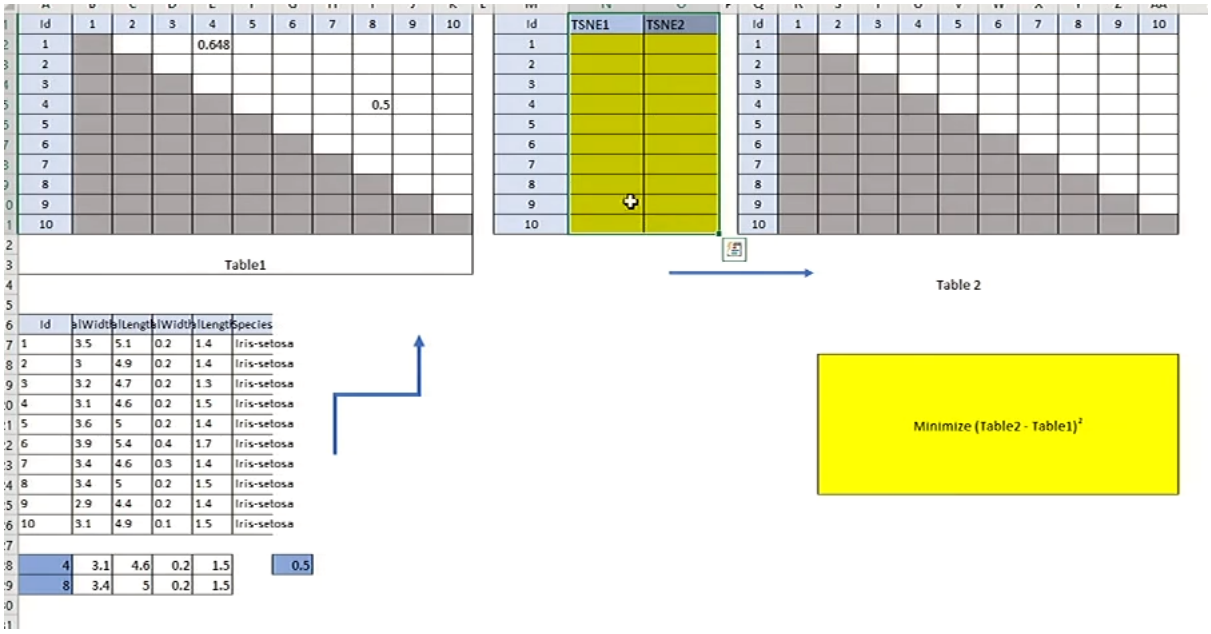
Id	SepalWidthCm	SepalLengthCm	PetalWidthCm	PetalLengthCm	Species
1	3.5	5.1	0.2	1.4	Iris-setosa
2	3	4.9	0.2	1.4	Iris-setosa
3	3.2	4.7	0.2	1.3	Iris-setosa
4	3.1	4.6	0.2	1.5	Iris-setosa
5	3.6	5	0.2	1.4	Iris-setosa
6	3.9	5.4	0.4	1.7	Iris-setosa
7	3.4	4.6	0.3	1.4	Iris-setosa
8	3.4	5	0.2	1.5	Iris-setosa
9	2.9	4.4	0.2	1.4	Iris-setosa
10	3.1	4.9	0.1	1.5	Iris-setosa

PCA1	PCA2
0.590855	0.319576
0.336288	0.458309
0.167849	0.897624
0.439823	0.404671

PCA1	PCA2
4.432382	4.201959
4.069697	3.950509
4.076628	3.882295
4.071878	3.885441
4.457838	4.188086
4.935127	4.768209
4.221937	4.030609
4.38365	4.164638
3.842467	3.689397
4.15598	3.933172

	SepalWidthCm	SepalLengthCm	PetalWidthCm	PetalLengthCm
SepalWidthCm				
SepalLengthCm			0.0078	
PetalWidthCm				
PetalLengthCm				

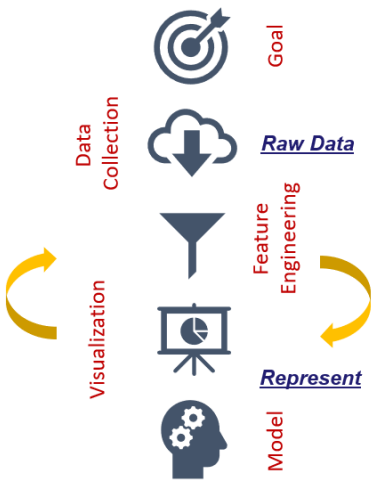
T-SNE:



We find Euclidean distance between two observations (similarity) and try to reduce it as much as possible.

Feature Engineering

- ✓ Feature Engineering is the key to improve the performance of Machine Learning models.
- ✓ Feature Engineering is the process of transforming raw data in the form that represent the problem in hand.
- ✓ In this course, following aspects related to feature engineering are covered
  - ✓ Data wrangling
  - ✓ Dimensions reduction and feature selection.
  - ✓ Preprocessing techniques and Data Visualization.
- ✓ Hands on experience



Feature Engineering – Why ??

Color / Shape	Red	Yellow	Green	Blue	Total
Triangle	12	4	5	7	28
Square	11	10	11	3	35
Circle	4	5	7	8	24
Total	27	19	23	18	87

Like					
Color / Shape	Red	Yellow	Green	Blue	Total
Triangle	4	2	3	2	11
Square	5	3	5	1	14
Circle	1	3	5	7	16
Total	10	8	13	10	41

Dislike					
Color / Shape	Red	Yellow	Green	Blue	Total
Triangle	8	2	2	5	17
Square	6	7	6	2	21
Circle	3	2	2	1	8
Total	17	11	10	8	46

- How many ***types*** of Toys are present?? How can I ***categorize***??
- Can I visualize all features of a Toy in a ***single chart***??

- Why someone
  - ***Likes / Dislikes***
  - ***Buys / Don't Buy***
- Can I ***predict*** above based on all or some features of a toy

- Which features, dimensions or measures, variables are ***important*** and not important?? in above two contexts

Color = 4
Shape = 3
*Total = 4*3 = 12*
*If I add 1 more dimension say complexity (simple, medium and complex), then the total gets 4*3*3 = 36*
(Curse of Dimensionality)

Feature Engineering – What ??

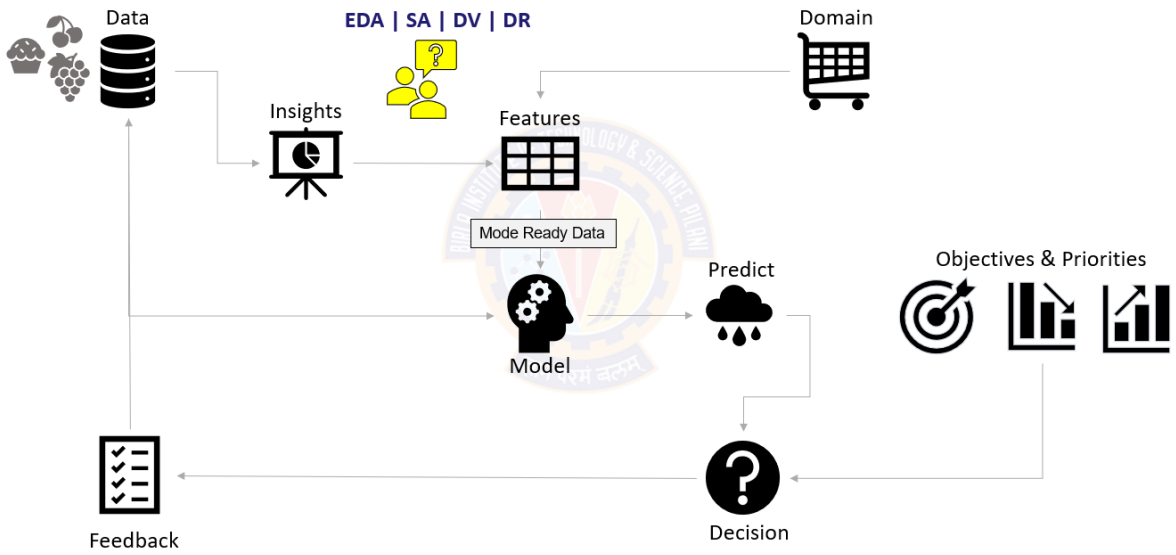
From Wikipedia, the free encyclopedia

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Feature engineering is fundamental to the application of machine learning, and is both difficult and expensive. The need for manual feature engineering can be obviated by automated feature learning.

Feature engineering is an informal topic, but it is considered essential in applied machine learning.

- ✓ In the current era of data science feature Engineering which is largely **DOMAIN** driven and time-consuming can be readily exploited using computational tools and data driven algorithms
- ✓ Following could be possible outcomes of feature engineering
  - ✓ You will come to know the importance of features IN A SPECIFIC CONTEXT
    (E.g. Temperature and Humidity are Vital for COMFORTABLE AMBIANCE)
  - ✓ Extract INDIRECT measures from DIRECT measures IN A SPECIFIC CONTEXT (CREDIT RISK)
    (E.g. Is Amount Spent on Your Credit Card Important or Percentage Balance Spent)
  - ✓ *Need for additional features*
  - ✓ *Understand 'Feature By itself should be proportional to Class'*

Feature Engineering – When ??



Feature Engineering – How ??

- Pre-Processing (E.g. Glare on Image is Not Important)
- Descriptive Statistics (E.g Avg. alone is Not Sufficient)
- Data Visualizations (E.g Understand Distribution)
- Transformation or Metric Definitions
  - (E.g X^2 may give better results than x itself)
  - (E.g $Amount\ Spend$ itself is not Important instead $Percent\ of\ Credit\ Card$ Spend is Important)
- Statistical Analysis (E.g. $Corr\ (X, Y)$ provides +/- relationship)
- Domain Knowledge (Very Important)
- Dimensional Reduction (E.g. A Photograph from mobile has > $10000\ Pixel\ Values$ i.e. dimensions)
- Surveys (E.g. Answers why $(i.e. which\ features)$ make them buy a specific product)