Machine Learning

Lecture 6-7: Data Preprocessing

COURSE CODE: CSE451

2023

Course Teacher

Dr. Mrinal Kanti Baowaly

Associate Professor

Department of Computer Science and Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Bangladesh.

Email: mkbaowaly@gmail.com

What is Data Preprocessing?

- A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models.
- Data preprocessing involves cleaning and transforming the data in a structured, useful and efficient format to make it suitable for analysis and machine learning models.
- It is the most important step in machine learning to ensure the quality of data.
- It increases the accuracy and efficiency of a machine learning model.

Data Preprocessing Techniques

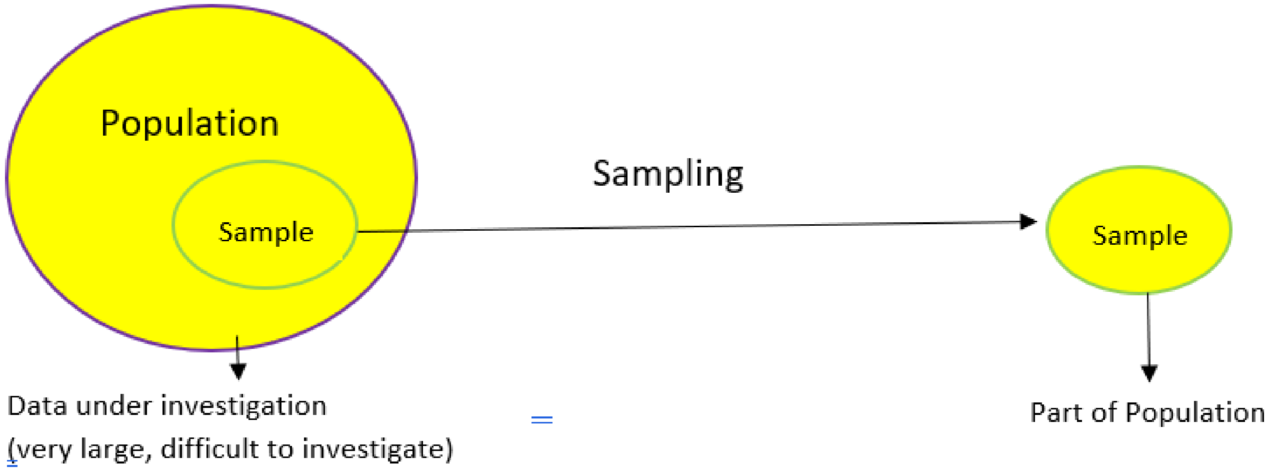
- Data cleaning (fix noises, outliers, missing values, duplicates in data)
- Aggregation
- Sampling
- Dimensionality reduction
- Feature subset selection
- Feature creation
- Discretization and binarization
- Attribute transformation

Aggregation

- Combining two or more attributes (or objects) into a single attribute (or object)
- Purpose
  - Data reduction
    - Reduce the number of attributes or objects
  - Change of scale
    - Cities aggregated into regions, states, countries, etc.
  - Less memory, less processing time
- Disadvantage: the potential loss of interesting details

Sampling

- Sampling is a method that allows us to get information about the population based on the statistics from a subset of the population (sample), without having to investigate every individual.
- Sampling is used in data mining because processing the entire set of data of interest is too expensive or time consuming.



Population

Sample

Sampling

Sample

Data under investigation
(very large, difficult to investigate)

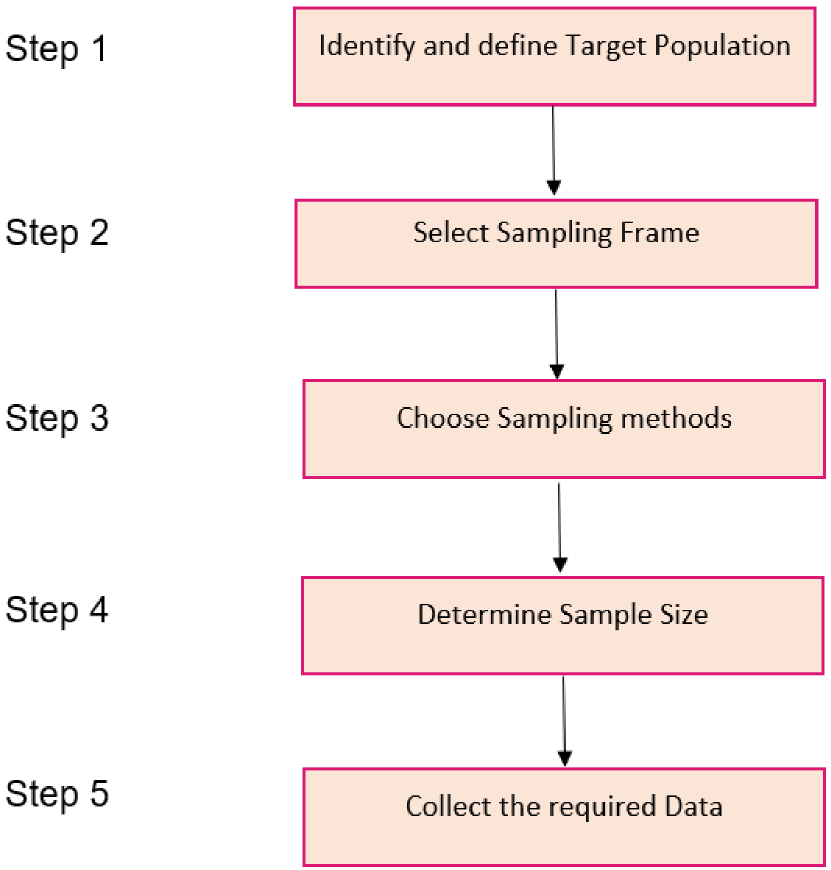
Part of Population

What is Representative Sample?

- The key principle for effective sampling is the following:
  - Using a sample will work almost as well as using the entire data sets (or population), if the sample is representative

  - A sample is representative if it has approximately the same property (of interest) as the original set of data

Steps Involved in Sampling



Step 1 Identify and define Target Population

Step 2 Select Sampling Frame

Step 3 Choose Sampling methods

Step 4 Determine Sample Size

Step 5 Collect the required Data

Case study: Public Opinion Polls

Target Population: People who are above 18 years and are eligible to vote.

Sampling Frame: It is a list of people forming a population from which the sample is taken, e.g., voter list.

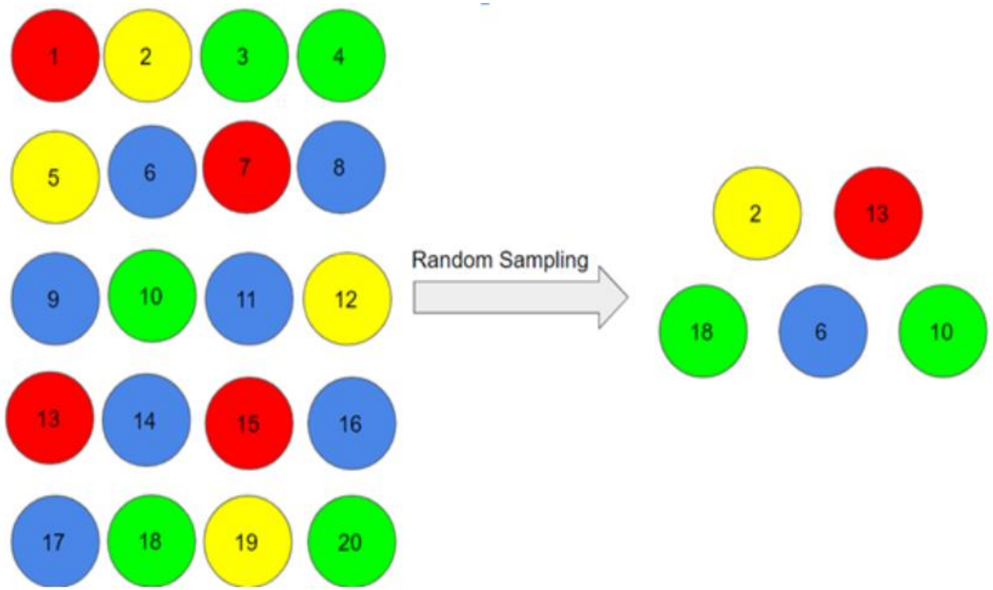
Detail: AanalyticsVidhya

Types of Sampling Methods

- Simple Random Sampling
- Systematic Sampling
- Stratified Sampling
- Cluster Sampling
- Multistage sampling

Simple Random Sampling

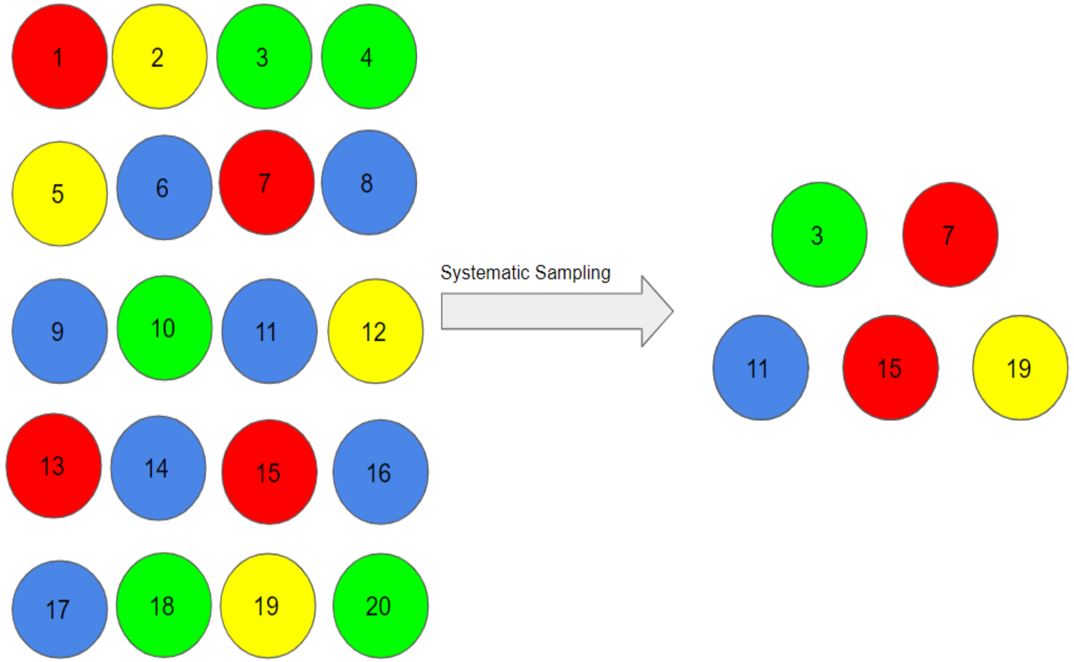
- Select a subset of items randomly from a population
- There is an equal probability of selecting any particular item
  - Sampling without replacement
    - As each item is selected, it is removed from the population
  - Sampling with replacement
    - Objects are not removed from the population as they are selected for the sample
    - In sampling with replacement, the same object can be picked up more than once



Systematic Sampling

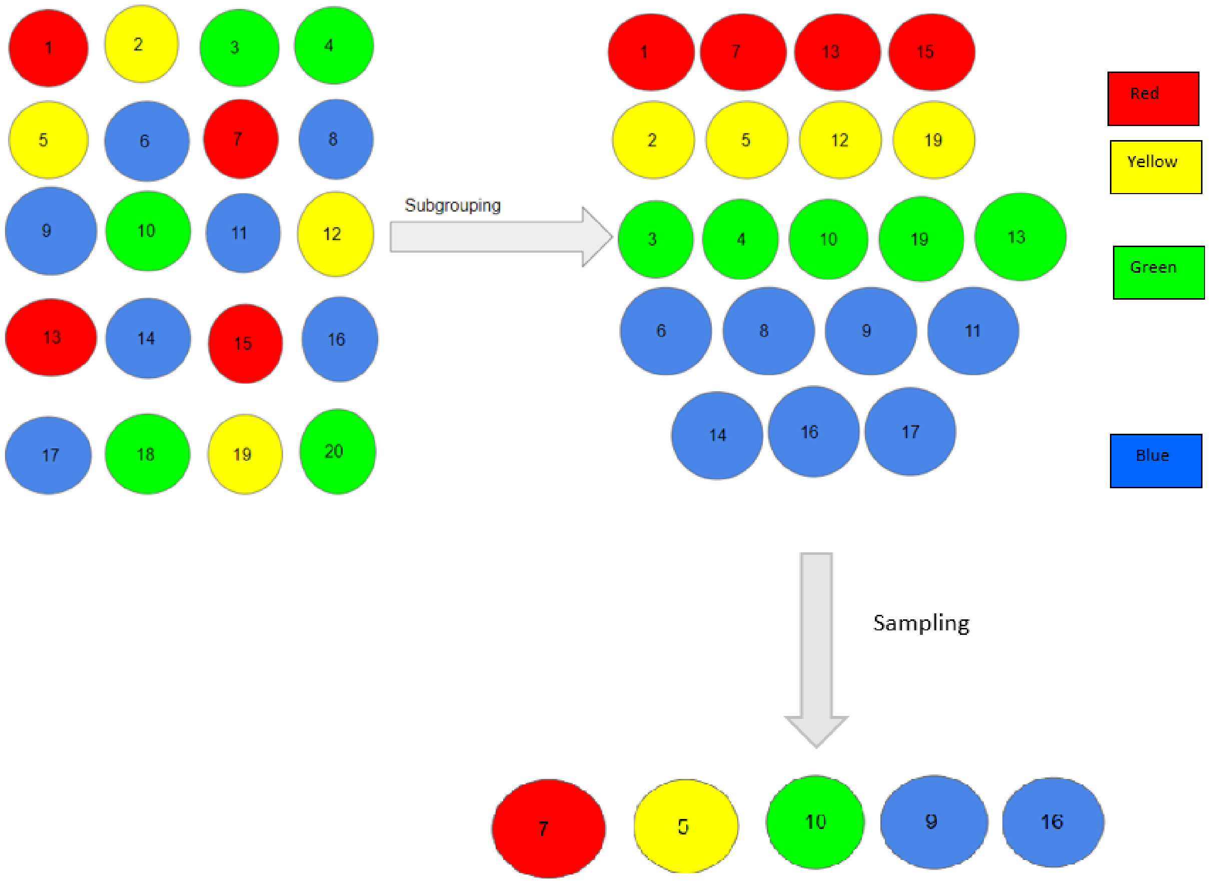
- Samples are drawn using a pre-specified pattern, such as at intervals
- Suppose, we began with person number 3, and we want a sample size of 5. So, the next individual that we will select would be at an interval of (20/5) = 4 from the 3rd person, i.e. 7 (3+4), and so on:

3, 3+4=7, 7+4=11, 11+4=15, 15+4=19



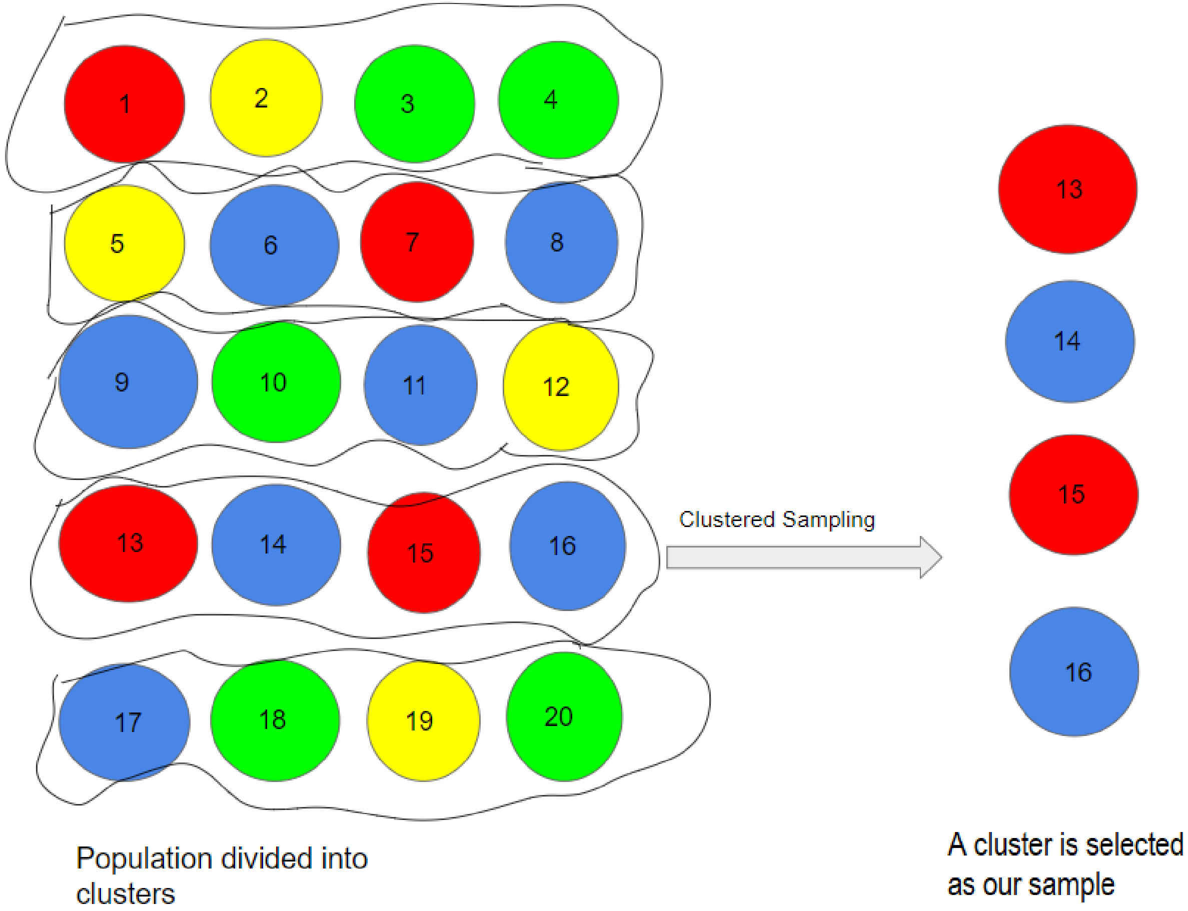
Stratified Sampling

- Split the data into several partitions called strata based on different traits like gender, category, etc.
- then draw random samples from each partition.



Cluster Sampling

- The population is divided into some groups called clusters.
- Then we select a fixed number of clusters randomly and include all observations from each of the clusters in our sample.
- In the example, one cluster is selected as our sample but we can include more clusters as per our sample size.

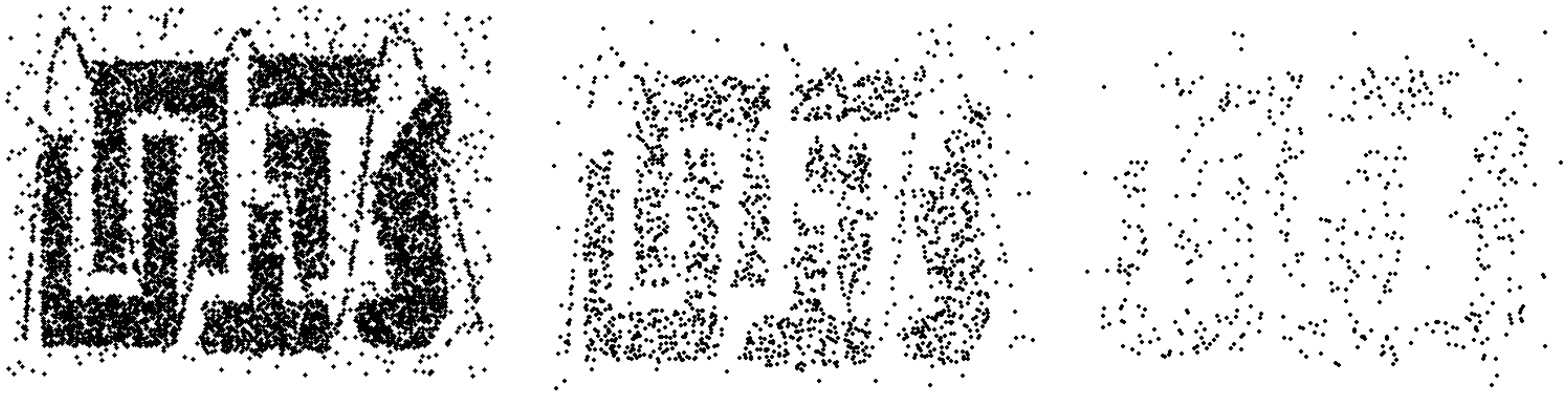

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Clustered Sampling
13
14
15
16
Population divided into clusters
A cluster is selected as our sample

Multistage Sampling

- **Multistage sampling:** It is very much similar to cluster sampling but instead of keeping all the observations in each cluster, we collect a random sample within each selected cluster.

Detail: AanalyticsVidhya, Kaggle

Determine the Proper Sample Size



8000 points

2000 Points

500 Points

Example of the loss of structure with sampling

- **Progressive sampling:** Start with a small sample, and then increase the size until a sufficient sample has been obtained

Curse of Dimensionality

- Many types of data analysis become harder as the dimensionality increases, the data becomes increasingly sparse in the space that it occupies

- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful

Dimensionality Reduction

Purpose:

- Avoid curse of dimensionality
- May help to eliminate irrelevant features or reduce noise
- Reduce amount of time and memory required by data mining algorithms
- Allow data to be more easily visualized
- Allow model to be more understandable

Techniques:

- Principle Component Analysis (PCA)
- Singular Value Decomposition (SVD)
- Others: supervised and non-linear techniques

Dimensionality Reduction vs Feature Subset Selection

Dimensionality Reduction

- Techniques that reduce the dimensionality of a data set by creating new attributes that are a combination of the old attributes

Feature (Subset) Selection

- Techniques that reduce the dimensionality of a data set by selecting only a subset of the attributes

Feature Selection

- Alternative way to reduce dimensionality of data.
- It is desirable to reduce the number of **redundant** and **irrelevant** input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model. It also reduces overfitting.
- Redundant features
  - Duplicate much or all of the information contained in one or more other attributes
  - Redundant features add no relevant information to your other features, because they are correlated or because they can be obtained by [linear] combination of other features.
  - Example: date of birth of a student and his age, age can be obtained from date of birth

Feature Selection (Cont.)

- Irrelevant features
  - Contain no information that is useful for the data mining task at hand
  - Example: students’ ID is often irrelevant to the task of predicting their GPA

Feature Selection Techniques

- Filter approaches/Univariate Statistics:
  - Features are selected using some statistical measures, e.g. Pearson’s Correlation, LDA, ANOVA, Chi-Square test etc.
  - We select a feature if there is a statistically significant relationship between the feature and the target.
  - These methods are generally used while doing the pre-processing i.e. before building a model.
  - Each feature is considered individually which can sometimes help when features are in isolation (don’t have a dependency on other features).

Feature Selection Techniques(Cont.)

- Embedded methods/Model-Based/Intrinsic
  - It uses a supervised machine learning model to judge the importance of each feature, and keeps only the most important ones.
  - More important features are assigned a higher weight, while less important features are given a lower weight.
  - Some machine learning algorithms (e.g. Decision trees, SVM, GBM) perform automatic feature selection during model training.
  - Regularization algorithms (e.g. LASSO, Elastic Net and Ridge Regression) are also used to do feature selection automatically.
  - In contrast to univariate selection, model-based selection considers all features at once, and so can capture interactions between features.

Feature Selection Techniques(Cont.)

- Wrapper approaches/Iterative Feature Selection:
  - A series of models are built, with varying numbers of features in an iterative manner.
  - Like greedy algorithm, it finds the optimal subset of features by evaluating all the possible subset of features against the model evaluation criterion.
  - The machine learning algorithm is used as a black box during model training for each subset of features.
  - Some common examples of wrapper methods are **forward feature selection**, **backward feature elimination**, **recursive feature elimination (RFE)**, etc.

Wrapper approaches/Iterative Feature Selection

- Forward Feature Selection:

  We start with an empty set of features in the model. In each iteration, we keep adding a feature which best improves our model till an addition of a new feature does not improve the performance of the model.

- Backward Feature Elimination:

  We start with all the features and removes the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features.

  The backward selection method works on removing variable iteratively on the basis of p-value. If a p-value of a variable is less than or equal to your significance level (typically ≤ 0.05), it is statistically significant.

Wrapper approaches/Iterative Feature Selection (Cont.)

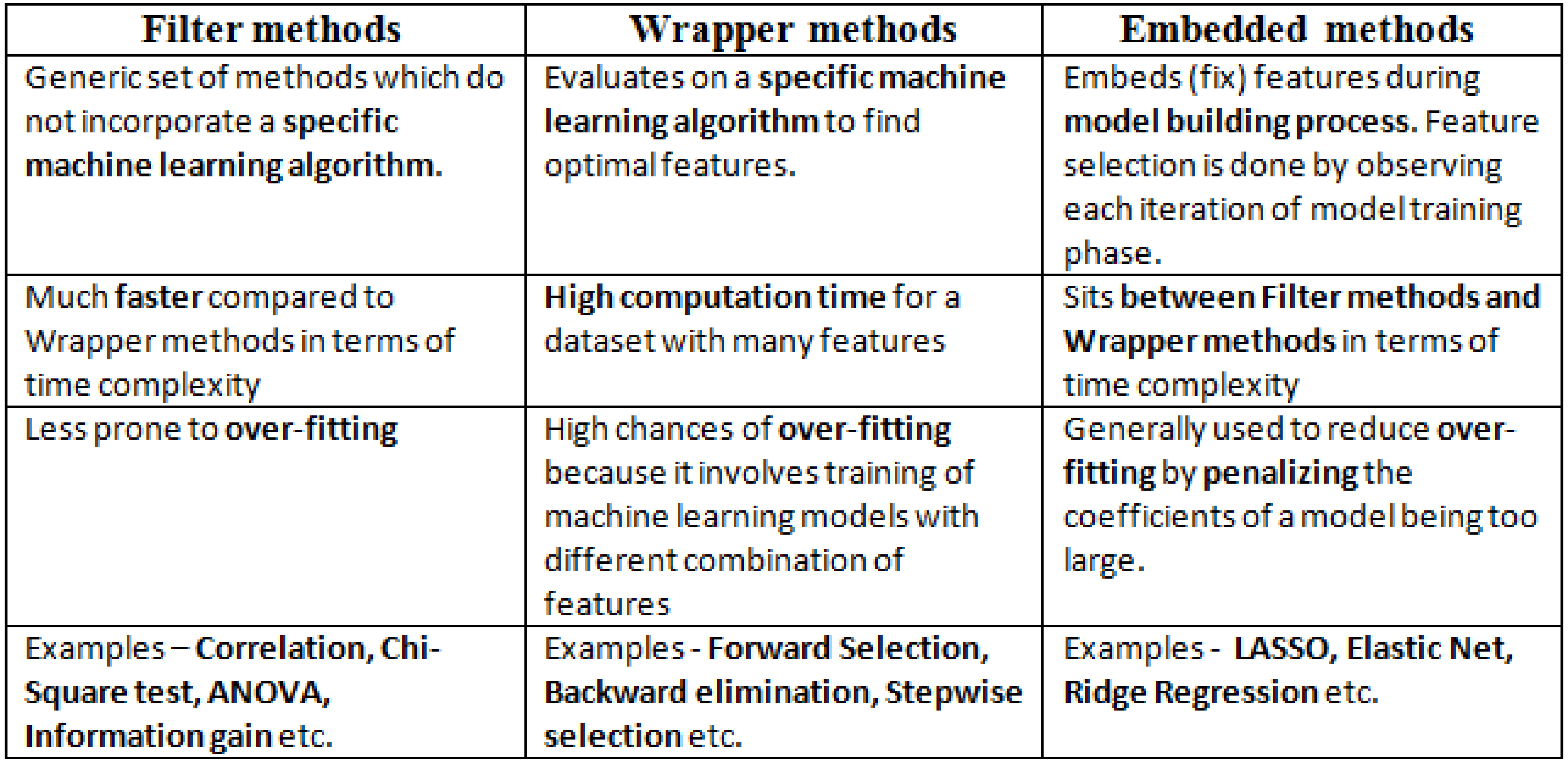
- Recursive Feature Elimination (RFE):

  RFE is also a type of backward selection method however RFE works on feature ranking system (e.g., based on the coefficients of a linear model) .

  RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains (say, 100).

  This is achieved by training the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-training the model. This process is repeated until a specified number of features remains.

Filter vs. Wrapper vs. Embedded Methods of Feature Selection



Filter methods	Wrapper methods	Embedded methods
Generic set of methods which do not incorporate a **specific machine learning algorithm.**	Evaluates on a **specific machine learning algorithm** to find optimal features.	Embeds (fix) features during **model building process.** Feature selection is done by observing each iteration of model training phase.
Much **faster** compared to Wrapper methods in terms of time complexity	**High computation time** for a dataset with many features	Sits **between Filter methods and Wrapper methods** in terms of time complexity
Less prone to **over-fitting**	High chances of **over-fitting** because it involves training of machine learning models with different combination of features	Generally used to reduce **over-fitting** by **penalizing** the coefficients of a model being too large.
Examples – **Correlation, Chi-Square test, ANOVA, Information gain** etc.	Examples - **Forward Selection, Backward elimination, Stepwise selection** etc.	Examples - **LASSO, Elastic Net, Ridge Regression** etc.

Feature Creation

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

Three general methodologies:

- Feature extraction
  - Example: extracting edges from images, word embedding or text vectorization
- Feature construction
  - Example: dividing mass by volume to get density
- Mapping data to new space
  - Example: Fourier and wavelet analysis

Discretization

Discretization is the process of converting a continuous attribute into an ordinal attribute

- A potentially infinite number of values are mapped into a small number of categories
- Discretization is commonly used in classification
- Many classification algorithms work best if both the independent and dependent variables have only a few values

How can we tell what the best discretization is?

- Unsupervised discretization: find breaks in the data values without using the class label information
  - Common approaches: Equal width, Equal frequency, K-means clustering

- Supervised discretization: Use class label information to find breaks i.e. supervised discretization filter uses the number of classes as the discretization parameter

Binarization

- Binarization maps a continuous or categorical attribute into one or more binary variables
- Many machine learning models (e.g. regression or SVM) and deep learning models require all input and output variables to be numeric. This means that if your data contains categorical data, you must encode it to numbers before you can fit and evaluate a model.
- Often convert a continuous attribute to a categorical attribute and then convert a categorical attribute to a set of binary attributes.

Encoding Categorical Attributes

- Label Encoding or Ordinal Encoding
- Binary Encoding
- One-hot Encoding

Source: AnalyticsVidhya, TowardsDatascience

Label Encoding or Ordinal Encoding

- We use this encoding technique when the categorical features are ordinal. In this case, retaining the order is important.
- In Label encoding, for N categories in a variable it uses N unique integers in the range [0, N-1].

Degree	Integer Value
Primary	0
Secondary	1
Higher Secondary	2
B.Sc.	3
M.Sc.	4
Ph.D.	5

Binary Encoding

- In this encoding scheme, the categorical feature is first converted into numerical using ordinal integer numbers. Then the numbers are transformed in the binary number.
- Each binary digit creates one feature column.

Color	Integer Value	X_0	X_1
Red	0	0	0
Green	1	0	1
Yellow	2	1	0

Data

Color
Red
Yellow
Red
Green
Yellow

Encoded Data

X_0	X_1
0	0
1	0
0	0
0	1
1	0

One-hot Encoding (Dummy Variables)

- One-hot encoding is essentially the representation of nominal categorical variables as binary vectors.
- The categorical values are first mapped to integer values. Each integer value is then represented as a binary vector that contains 1 and 0 denoting the presence or absence of the feature.

Color	Integer Value	Red	Green	Yellow
Red	0	1	0	0
Green	1	0	1	0
Yellow	2	0	0	1

✓ The newly created binary features (dummy variables) depends on the categories present in the variable

Data

Color
Red
Yellow
Red
Green
Yellow

Encoded Data

Red	Green	Yellow
1	0	0
0	0	1
1	0	0
0	1	0
0	0	1

Variable/Attribute Transformation

An attribute transform is a function that maps the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values

- Simple functions: x^k, log(x), e^x, $|x|$, sqrt, sin x, 1/x etc.
- **Purpose**: sqrt, log and 1/x are often used to transform data to Gaussian (normal) distribution, minimizing the huge range of values

Math: Normal distribution, Standard Deviation

Normalization

Normalization scales all numeric variables in the range [0,1]

- Refers to various techniques to adjust the differences among attributes in terms of frequency of occurrence, mean, variance, range
- Before normalization, it is recommended to handle the outliers

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Detail: How to Normalize Data in Python

Normalization Example

```
# Normalize the data attributes for the Iris dataset.
from sklearn.datasets import load_iris
from sklearn import preprocessing
# load the iris dataset
iris = load_iris()
print(iris.data.shape)
# separate the data from the target attributes
X = iris.data
y = iris.target
# normalize the data attributes
normalized_X = preprocessing.normalize(X)
```

Standardization

Data standardization is the process of rescaling one or more variables so that they have a mean value of 0 and a standard deviation of 1

- Refers to subtracting off the means and dividing by the standard deviation
- Useful when min and max are unknown or when there are outliers

$$x_{new} = \frac{x - \mu}{\sigma}$$

Detail: Standardize Data in Python, Rescaling Data for Machine Learning

Standardization Example

```
# Standardize the data attributes for the Iris dataset.
from sklearn.datasets import load_iris
from sklearn import preprocessing
# load the Iris dataset
iris = load_iris()
print(iris.data.shape)
# separate the data and target attributes
X = iris.data
y = iris.target
# standardize the data attributes
standardized_X = preprocessing.scale(X)
```

Exploratory Data Analysis (EDA)

- An approach to analyze and investigate data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods.
- EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task.
- It can help to identify obvious errors, as well as better understand the patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

Common techniques and tools of EDA

- Box plot: Link1, Link2
- Histogram: Link1, Link2
- Scatter plot
- Pairplot
- Heat map or pairwise correlation
- etc.

EDA Lab Works

1. Ultimate guide for Data Exploration in Python using NumPy, Matplotlib and Pandas, by AnalyticsVidhya
2. Introduction to Exploratory Data Analysis (EDA), by AnalyticsVidhya
3. Comprehensive Data Exploration with Python, by Kaggle
4. CheatSheet: Data Exploration using Pandas in Python, by AnalyticsVidhya
5. Python Exploratory Data Analysis Tutorial, by Datacamp
6. Statistical Learning Tutorial for Beginners, by Kaggle
7. Course: Pandas for Data Analysis in Python, by AnalyticsVidhya

End of Lecture-6,7