JAIN DEEMED-TO-BE UNIVERSITY | SCHOOL OF COMPUTER SCIENCE AND IT

School of Computer Science & IT

BCA Programme

INTRODUCTION TO DATA ANALYTICS (23BCAD4C01)

MODULE 2: Data Preparation

Dr. Ananta Charan Ojha, Professor

Session -1

- Data Collection Methods
- Data Pre-processing

Data Collection Methods

- Data collection is the cornerstone of any successful data science or data analyst project.
- It is a process of gathering information from all the relevant sources to find a solution to the problem using data analytics. It helps to analyse the context and understand the problem.
- The data used for analysis can be of two types: Primary and Secondary.
- **Primary Data**: Data that has been generated by the data scientist/ analyst himself/herself, using surveys, interviews, experiments, specially designed for understanding and solving the problem at hand.
- **Secondary Data**: Data that has already been collected by someone else.
  - It may be available in public repositories like Kaggle, UCI Machine Learning repository; Publicly available datasets: Census data, weather data, economic indicators.
  - Internal company data: Sales records, customer databases, financial reports existing in organizational Information Systems as part of organizational record keeping. The data is then extracted from more varied datafiles.
  - The secondary data also includes magazines, newspapers, books, journals, etc. It may be either published data or unpublished data.
- Depending on the type of data, the data collection method is divided into two categories namely,
  - Primary Data Collection methods
  - Secondary Data Collection methods

Primary Data Collection

- **Questionnaire Method:** Questionnaires are a simple, straightforward data collection method. Respondents get a series of questions, either open or close-ended, related to the matter at hand. They should read, reply and subsequently return the questionnaire. They can be administered online, via mail, or in person.
- **Interview Method:** The researcher asks questions of a large sampling of people, either by direct interviews or means of mass communication such as by phone or mail. This method is by far the most common means of data gathering.
  - **Personal Interview** – In this method, a person known as an interviewer is required to ask questions face to face to the other person. The personal interview can be structured or unstructured, direct investigation, focused conversation, etc.
  - **Telephonic Interview** – In this method, an interviewer obtains information by contacting people on the telephone to ask the questions or views orally.
- **Observation Method:** Observation method is used when the study relates to behavioural science. This method is planned systematically. It is subject to many controls and checks. The different types of observations are: participant observation (where the researcher takes part in the activities), and onlooker observation (where the researcher simply observes the activities).
- **Focus Groups:** Focus groups, like interviews, are a commonly used technique. The group consists of anywhere from a half-dozen to a dozen people, led by a moderator, brought together to discuss the issue.

Secondary Data Collection

- Unlike primary data collection, there are no specific collection methods.
- Since secondary data has varieties of sources published / unpublished documents/records, web pages to organizational databases, Secondary Data collection methods can be from manual process to automated ones using programming.
- In organizations, data warehouses are home to all transactional data.
  - Still the possibility exists that your data may reside in Excel files on the desktop of a domain expert.
- Finding data even within your own company can sometimes be a challenge.
  - As companies grow, their data becomes scattered around many places. Knowledge of the data may be dispersed as people change positions and leave the company.
- Getting access to data is another difficult task.
  - Organizations understand the value and sensitivity of data and often have policies in place so everyone has access to what they need and nothing more.
  - These policies translate into physical and digital barriers called *Chinese walls*.

Example Scenario: Analyzing Customer Churn

- Imagine a telecommunications company wants to understand why customers are leaving their service.
- Primary Data:
  - Conduct surveys with departing customers to ask about their reasons for leaving.
  - Analyze call logs to identify common customer complaints.
- Secondary Data:
  - Use customer demographic data (age, location, income) from internal databases.
  - Compare pricing of competitors for similar services/products to identify potential external factors.

Why Data Pre-processing

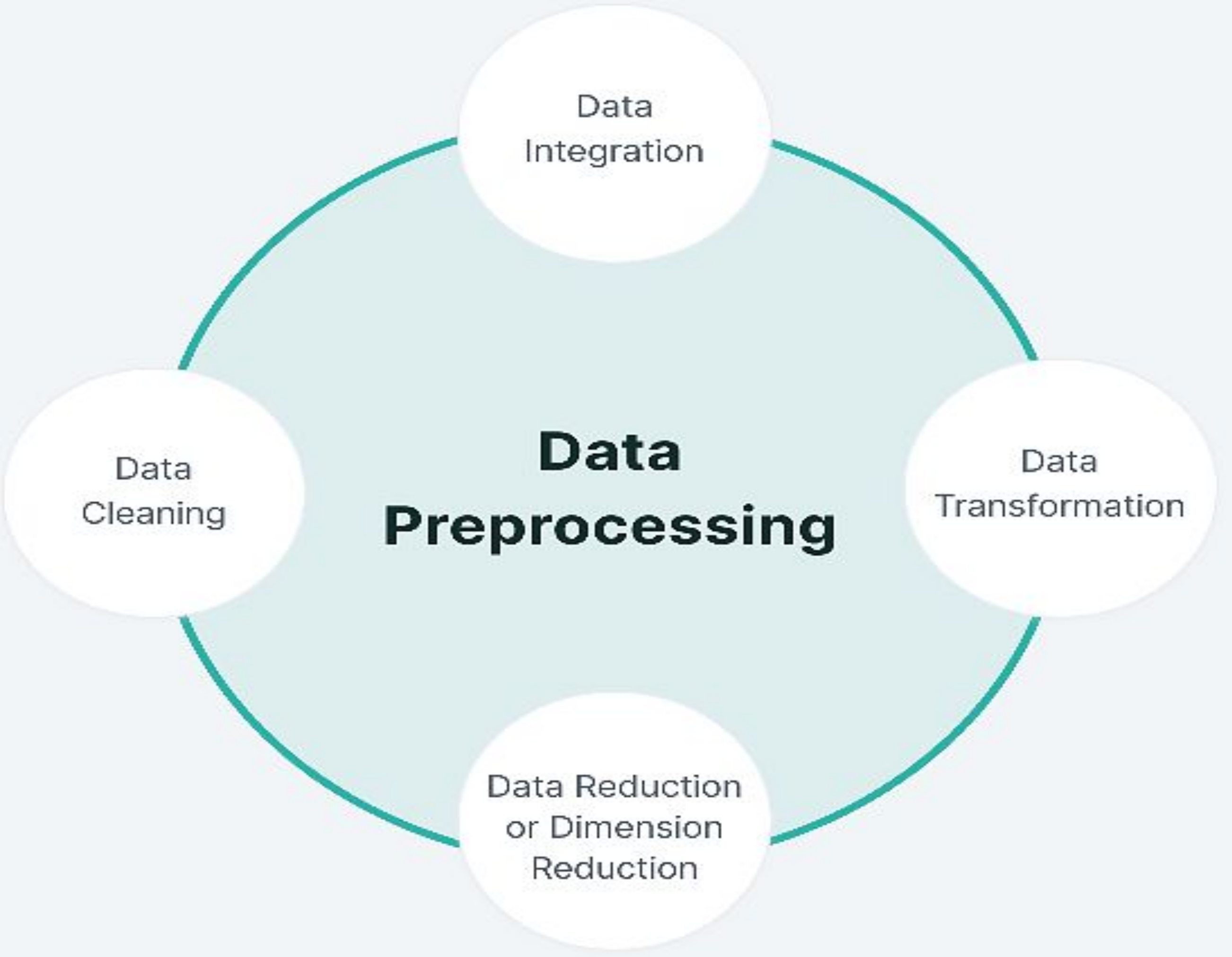
- Data pre-processing is the process of transforming raw data into machine understandable format.
- The majority of the real-world datasets are highly susceptible to missing, inconsistent, and noisy data due to their heterogeneous origin.
- Applying ML algorithms on this noisy data would not give quality results as they would fail to identify patterns effectively.
- The main goal of pre-processing is to improve the quality of data such that the machine learning algorithm should be able to easily interpret the data features, find the hidden pattern/knowledge and generate a model to be accurate and precise in prediction.
- Poor data quality can lead to inaccurate insights and produce unreliable results. Models built on flawed data will produce incorrect predictions.

Quality Factors

Data quality is paramount in data science as it directly impacts the accuracy and reliability of insights and models. Here are some key factors to consider:

- **Accuracy:** The degree to which data reflects the true value or state of the real-world entity it represents. Data must be correct. Outdated data, typo errors, and redundancies can affect a dataset's accuracy.
  - **Example**: A customer's actual address versus a misspelled or outdated address.
- **Consistency:** The degree to which data is uniform, follows standards, and is free from contradictions within and across datasets. The data should have no contradictions.
  - Example: Different date formats used for customer birthdates or inconsistent spellings of product names.
- **Completeness:** The degree to which all required data elements are present and have values. The dataset shouldn't have incomplete attributes or empty values.
  - **Example**: Missing values for age, income, or purchase history in a customer dataset.
- **Validity:** The degree to which data conforms to defined business rules and constraints. A dataset is considered valid if the data samples appear in the correct format, are within a specified range, and are of the right type. Invalid datasets are hard to organize and analyze.
  - **Example**: A customer's age being recorded as negative or a product price being listed as zero.
- **Timeliness:** The degree to which data is current and up-to-date for its intended use. Data should be collected as soon as the event it represents occurs. As time passes, every dataset becomes less accurate and useful as it doesn't represent the current reality. Therefore, the topicality and relevance of data is a critical data quality characteristic.
  - **Example**: Using outdated sales data for forecasting future trends.
- **Uniqueness**: The degree to which each record or entity is distinct and free from duplicates.
  - **Example**: Duplicate customer records with slightly different contact information.

Major Tasks in Data Pre-processing



❑ Data cleaning / cleansing:

- Data cleaning is the process of cleaning datasets by accounting for missing values, removing outliers, correcting inconsistent data points, and smoothing noisy data.
  - In essence, the motive behind data cleaning is to offer complete and accurate samples for machine learning models.

- **Missing values:** The problem of missing data values is quite common.
  - It may happen during data collection or due to some specific data validation rule. In such cases, you need to collect additional data samples or look for additional datasets.
  - The issue of missing values can also arise when you merge two or more datasets to form a bigger dataset. If all fields are not present/common in both datasets, it’s better to delete such fields before merging.

Handling Missing Values

- Here are some ways to account for missing data:
- **Data Deletion**: Remove entire rows (instance) containing missing values.
  - Can significantly reduce sample size, especially with multiple variables having missing data.
- **Data Imputation:**
  - **Manually fill in the missing values.**
    - This can be a tedious and time-consuming approach and is not recommended for large datasets.
  - **Make use of a standard value to replace the missing data value.**
    - You can use a global constant like "unknown" or "N/A" or default values to replace the missing value. Although a straightforward approach, it isn't foolproof.
  - **Fill the missing value with the most probable value.**
    - To predict the probable value, you can use algorithms like **logistic regression** or decision trees.
  - **Use a central tendency to replace the missing value.**
    - Central tendency is the tendency of a value to cluster around its mean, mode, or median.

Noisy data

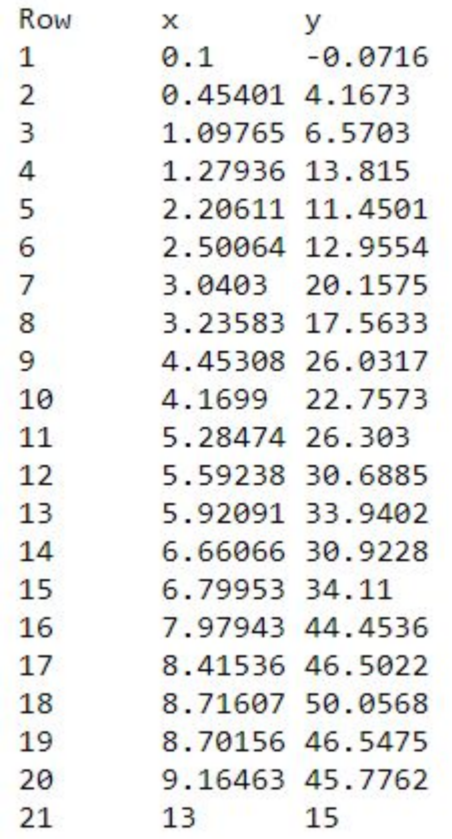
- Noisy data refers to data that contains errors, inaccuracies, or irrelevant information that can interfere with the analysis or interpretation of the data. This noise can come in various forms, such as:
- **Random Errors**: These are unpredictable mistakes that occur during data collection or processing. Examples include:
  - **Typos**: Incorrectly entered data, such as misspelled names or transposed numbers.
  - **Measurement** Errors: Inaccurate readings from sensors or instruments.
  - **Data Entry** Errors: Mistakes made during manual data entry.
- **Systematic Errors**: These are errors that occur consistently in a predictable way. Examples include:
  - **Bias**: A systematic tendency to favor certain outcomes over others.
  - **Calibration** Issues: Problems with the calibration of measuring instruments.
  - **Sampling** Bias: A non-representative sample that does not accurately reflect the population.
- Also, Noise includes
  - data having incorrect attribute values,
  - duplicate or semi-duplicates of data points,
  - data segments of no value
  - unwanted/irrelevant attributes.
- **Outliers**: Data points that are significantly different from the rest of the data. These can be caused by errors or represent legitimate but unusual observations.
- An outlier can be treated as noise, although some consider it a valid data point.
  - For numeric values, you can use a scatter plot or box plot to identify outliers.

Examples of Noisy Data

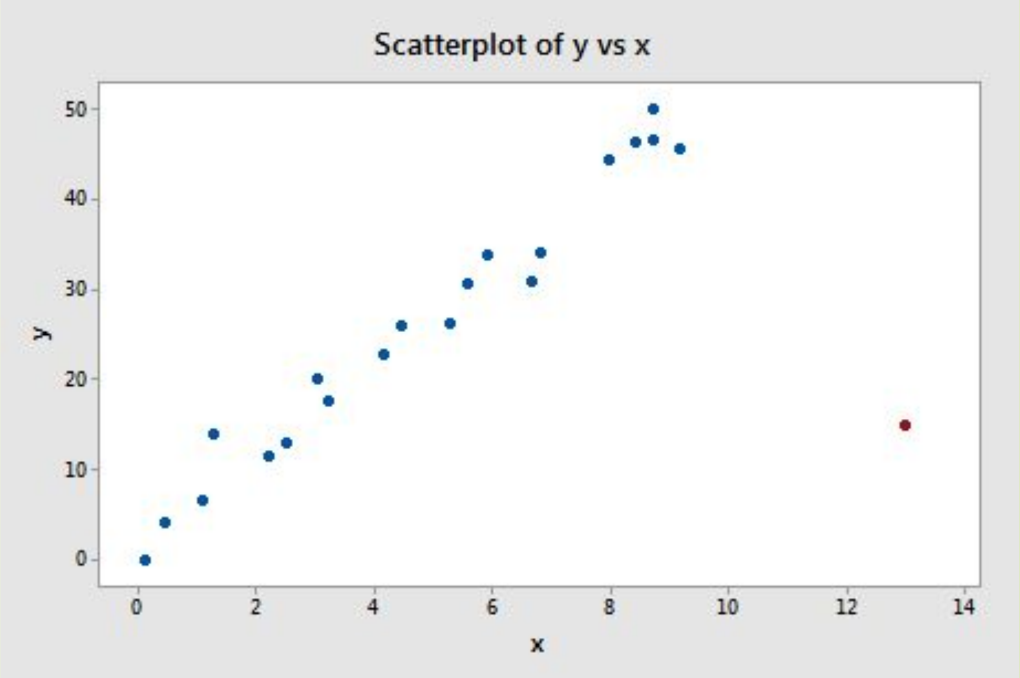
- A customer database with misspelled names and addresses.
- A medical dataset with inaccurate blood pressure readings due to faulty equipment.
- A survey with biased questions that lead respondents to give certain answers.
- A financial dataset with fraudulent transactions that distort the true picture of financial activity.
- A social media dataset with spam accounts and fake news that can skew the analysis of public opinion in sentiment analysis.

Outliers

- An outlier is a data point that is noticeably different from the rest. It is an unusual data point that does not follow the general trend of the rest of the data. It can be an extreme value.
- Outliers can arise for a variety of reasons, such as errors in measurement or bad data collection (e.g. Human error, Measuring Instrument error, Experiment design error etc.).
- Machine learning algorithms are sensitive to the range and distribution of attribute values.
- Data outliers can spoil and mislead the training process resulting in less accurate models and poorer results.



Row	x	y
1	0.1	-0.0716
2	0.45401	4.1673
3	1.09765	6.5703
4	1.27936	13.815
5	2.20611	11.4501
6	2.50064	12.9554
7	3.0403	20.1575
8	3.23583	17.5633
9	4.45308	26.0317
10	4.1699	22.7573
11	5.28474	26.303
12	5.59238	30.6885
13	5.92091	33.9402
14	6.66066	30.9228
15	6.79953	34.11
16	7.97943	44.4536
17	8.41536	46.5022
18	8.71607	50.0568
19	8.70156	46.5475
20	9.16463	45.7762
21	13	15



THANK YOU

Any questions…?

School of Computer Science & IT

BCA Programme

INTRODUCTION TO DATA ANALYTICS (23BCAD4C01)

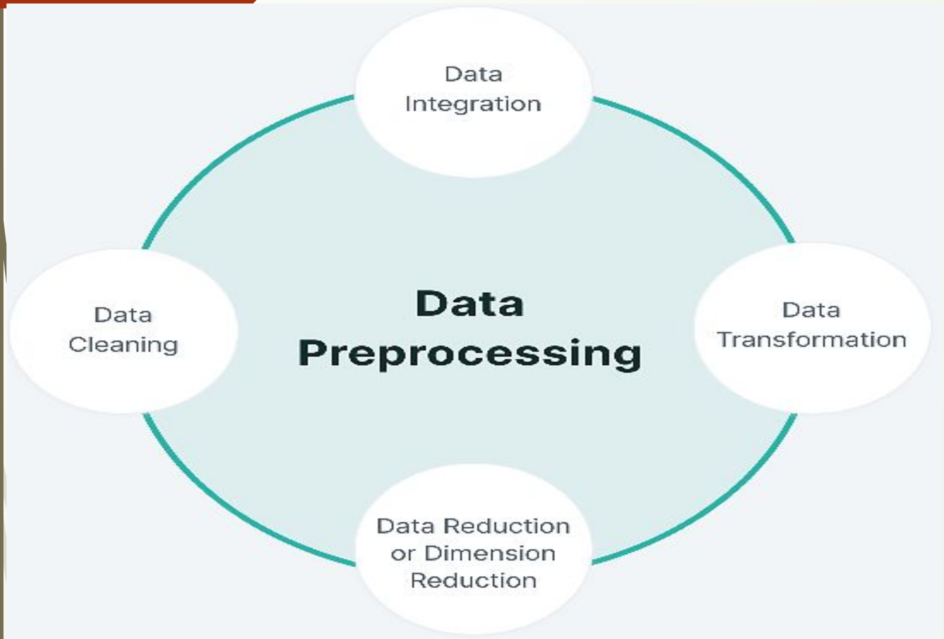
MODULE 2: Data Preparation

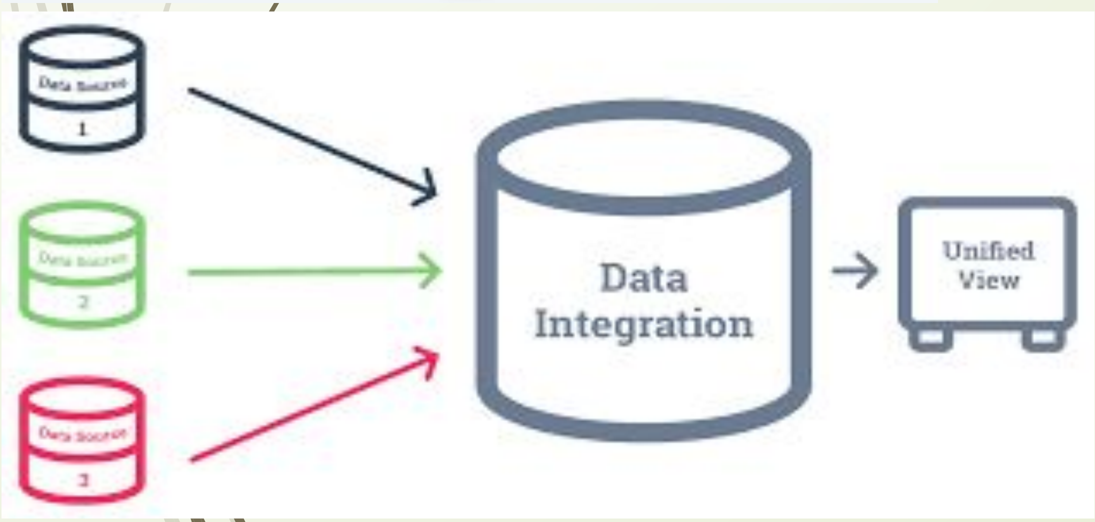
Dr. Ananta Charan Ojha, Professor

Session -2

- Data Pre-processing
  - Data Integration

Major Tasks in Data Pre-processing





❑ Data Integration

- Data integration is the process of combining/ merging data from different sources.
  - Example: A retail company might have customer data in its CRM system, sales data in its ERP system, and website browsing behavior in server logs, or a separate analytics platform. This isolated systems prevent a holistic view of customer’s interaction data.
- The data sources might be heterogenous in nature.
  - Data can come in various formats (CSV, JSON, XML), structures (relational, NoSQL), and granularities, making it challenging to integrate.
- These sources may include multiple data cubes (A data cube in a data warehouse is a multidimensional structure used to store data), databases, or flat files.
- Data integration results in a coherent/ well-organized data store and provides a unified view of the data.
- Combining data from a structured relational database with unstructured and semi-structured data requires significant transformation and normalization.

Data Integration Approaches

- Main approaches to integrate data:
- **Data consolidation:**
  - Data is physically brought together and stored in a single place.
  - Having all data in one place increases efficiency and productivity.
  - This step typically involves using data warehouse software (i.e. data integration tools: ETL).
  - **Extract, transform, and load** (ETL) technology supports data consolidation.
  - ETL pulls data from sources, transforms it into an understandable format, and then transfers it to another database or data warehouse.
  - The ETL process cleans, filters, and transforms data, and then applies business rules before data populates the new source.
- **Data virtualization:**
  - Virtualization uses an interface to provide a near real-time, unified view of data from disparate sources with different data models.
  - Data can be viewed in one location but is not stored in that single location.
  - Data virtualization retrieves and interprets data but does not require uniform formatting or a single point of access.

Issues in Data Integration

- There are some issues to be considered during data integration.

- **Schema Integration:**
  - Schema integration is the process of combining multiple schemas from different data sources into a unified schema. This is crucial for creating a consistent and coherent view of data across various systems, enabling seamless data sharing and analysis.
  - The schema integration can be achieved using metadata (a set of data that describes other data) of each attribute.
  - Metadata of an attribute incorporates its name,
    - what does it mean in the particular scenario,
    - what is its data type, up to what range it can accept the value.
    - What rules does the attribute follow for the null value, blank, or zero?
  - Analyzing this metadata information will prevent error in schema integration.
  - **Entity identification problem:** (Name conflict, attribute mismatch)
  - The real-world entities from multiple sources need to be matched correctly.
  - For example:
    - we have customer data from two different data source. An entity from one data source has customer_id and the entity from the other data source has customer_number.
    - Similarly, Semantic Heterogeneity: "order_status" might have different meanings in CRM (e.g., "pending," "shipped," "canceled") and ERP (e.g., "processing," "backordered," "delivered").
  - Here, the data analyst or the integration tool needs to understand that these two entities refer to the same attribute.

- **Redundancy (or Attribute Redundancy):**
  - Redundancy is one of the big issues during data integration.
  - Redundant data is an unimportant data or the data that is no longer needed.
  - An attribute may be redundant if it can be derived or obtaining from another attribute or set of attributes.
  - For example, one data set has the customer age and other data set has the customers date of birth then age would be a redundant attribute as it could be derived using the date of birth.
- Some redundancies can be detected by correlation analysis. The attributes are analyzed to detect their interdependency on each other thereby detecting the correlation between them – collinearity.
- **Collinearity**:
  - *Collinearity* refers to the situation in which two or more predictor variables are closely related to one another.
  - The presence of collinearity can pose problems, since it can be difficult to separate out the individual effects of collinear variables on the response variable.
  - When faced with the problem of collinearity, there are two simple solutions.
    - The **first** is to drop one of the problematic variables from the data. This can usually be done without much compromise to the prediction, since the presence of collinearity implies that the information that this variable provides about the response is redundant in the presence of the other variables.
    - The **second** solution is to combine the collinear variables together into a single predictor. For instance, we might take the average of standardized versions of there variables in order to create a new variable.

Example: House Price Prediction - Predictor Variables:

- Size of the house (square footage)
- Number of bedrooms
- Number of bathrooms

Collinearity: Larger houses typically have more bedrooms and bathrooms, leading to a strong correlation between these variables.

- **Data Value Conflict:**
  - Data conflict means the data merged from the different sources do not match.
  - Attribute values from different sources may differ for the same real-world entity. The difference maybe because they are represented differently in the different data sets.
  - For example, the price of a hotel room may be represented in different currencies in different cities. Likewise, the date format may differ like “MM/DD/YYYY” or “DD/MM/YYYY”.
  - Detection and resolution of data value conflicts need to be carried out carefully.

- **Tuple Duplication:**
  - Along with redundancies, data integration has to deal with the duplicate tuples/ rows also.
  - Duplicate tuples may come in the resultant data if the denormalized table has been used as a source for data integration.
  - Database normalization is **used to remove redundant data** from the database and to store non-redundant and consistent data into it.

- Also, appropriate SQL Clauses may be used to extract unique tuples in data

```
SELECT DISTINCT column1, column2, ...
FROM table_name;


SELECT column1, column2, ...
FROM table_name
GROUP BY column1, column2, ...;
```

THANK YOU

Any questions…?

School of Computer Science & IT

BCA Programme

INTRODUCTION TO DATA ANALYTICS (23BCAD4C01)

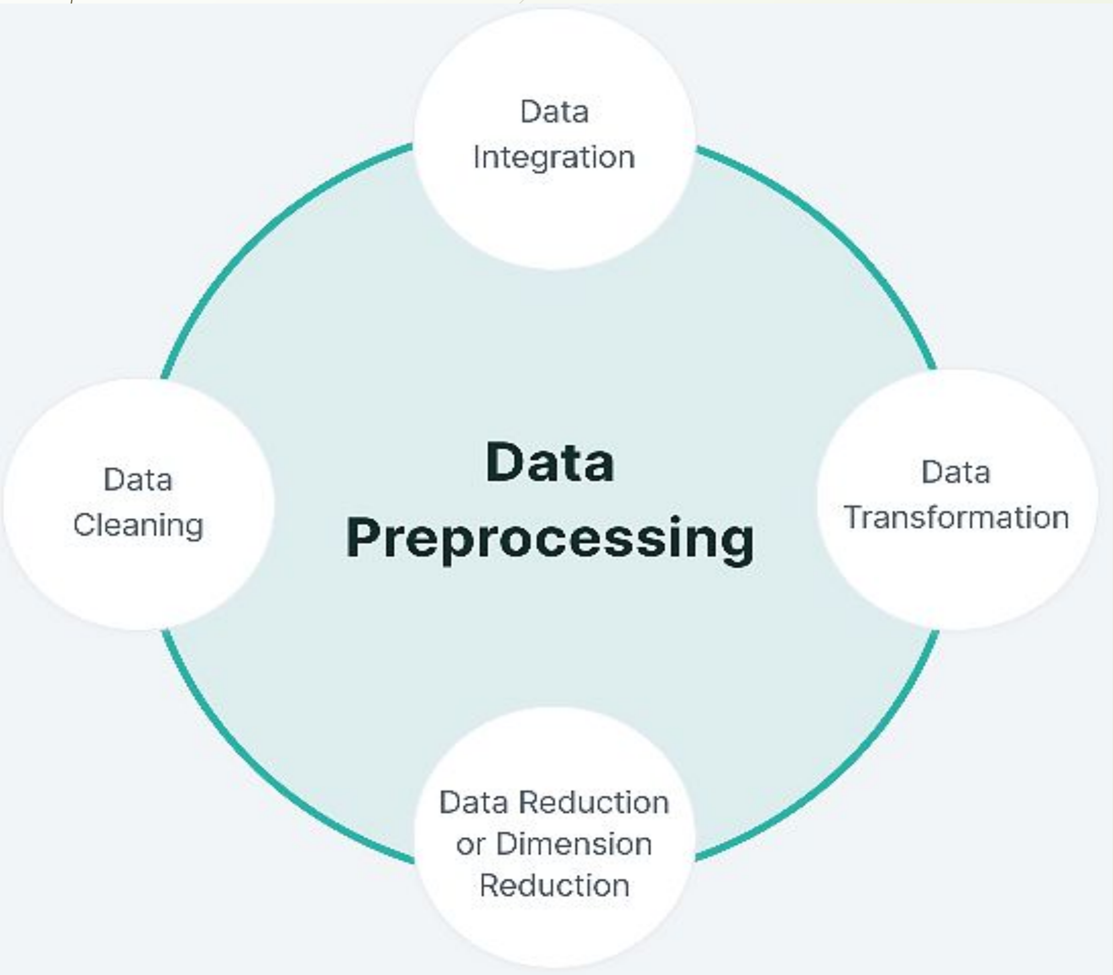
MODULE 2: Data Preparation

Dr. Ananta Charan Ojha, Professor

Session -3

- Data Pre-processing
  - Data Transformation

Major Tasks in Data Pre-processing



- **Data Transformation**
  - Data transformation is the process of converting data from one format to another.
  - In essence, it involves methods for transforming data into appropriate formats that the computer can learn efficiently from.
  - Data transformation increases the **efficiency of** analytic processes, and it enables businesses to make better data-driven decisions.
  - Data transformation could change the structure, format, or values of data by
    - Moving, renaming, and combining columns in a database.
    - Adding, copying, and replicating data.
    - Deleting records and columns in a database
  - Typically, data engineers, data scientists, and data analysts use programming languages such as SQL or Python to transform data.
  - Organizations may also choose to use **ETL (Extract, Transform Load) tools**, which can automate the data transformation process for data stored in different databases.
  - Different tools provide different features for data transformation.
  - This step can be simple or complex based on the requirements.

Methods in Data Transformation

Data aggregation

- Data aggregation is the process where raw data is gathered and expressed in a summary form for statistical analysis.
  - **For example**, raw data can be aggregated over a given time period to provide statistics such as average, minimum, maximum, sum, and count. After the data is aggregated and written to a view or report, you can analyze the aggregated data to gain insights about particular resources or resource groups.
- There are two types of data aggregation:
  - **Time aggregation**
    - All data points for a single resource over a specified time period.
  - **Spatial aggregation**
    - All data points for a group of resources over a specified time period. Example: Vegetable, Grocery, Garments sales in a super bazar.

- Data is collected and presented in a view within the context of various time intervals:
- **Reporting period:** The period over which data is collected for presentation. For example, Daily, Weekly, Monthly, Quarterly, and Yearly.
- **Polling period:** The time duration that determines how often resources are sampled for data. For example, a group of resources might be polled every 5 minutes, meaning that a data point for each resource is generated every 5 minutes.

❖ **Example**: Weather forecasting, Sensor data for temperature, humidity polled in every 5 minutes, aggregated to minimum and maximum and reported daily. Daily sales of a super bazar store aggregated for monthly average forecasting

Data Smoothing

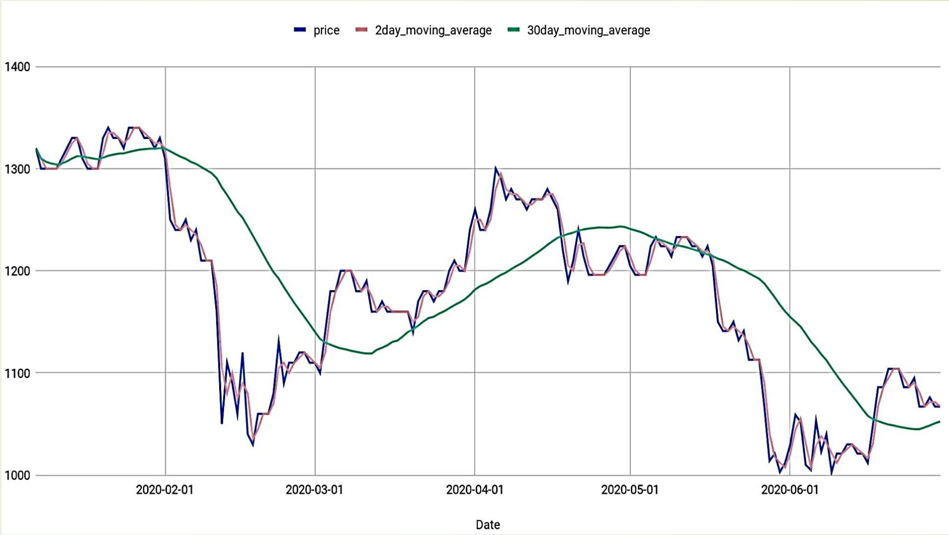
- Data smoothing is statistical techniques performed to remove noise from a data set. This allows important patterns to more clearly stand out.
- Data collected over time displays random variation; smoothing techniques can be used to reduce or cancel the effect of these variations.
- When properly applied, these techniques smooth out the random variation in the time series data to reveal underlying trends. (Example of Time Series Data: Daily max, min price of a stock value over a month)
  - **For example**, Data smoothing can be used to help predict trends, such as those found in securities prices, as well as in economic analysis.
- Data Analytics Tools features four different smoothing techniques: Exponential, Moving Average, Double Exponential, and Holt-Winters.
- Moving Average: This method calculates the average of a specific number of data points over a certain time period. The average is then used as the smoothed value for the middle data point in the window.
- Exponential Smoothing: This method assigns more weight to recent data points than older ones, making it more responsive to recent changes.
  - Exponential and Moving Average are relatively simple smoothing techniques and should not be performed on data sets involving seasonality.
  - Double Exponential and Holt-Winters are more advanced techniques that can be used on data sets involving seasonality.
- Seasonality is a characteristic of a time series in which the data experiences regular and predictable changes that recur or repeat every calendar year. Any predictable fluctuation or pattern that recurs or repeats over a one-year period is said to be seasonal.

Example of Moving Average

Let's say we have the following daily stock prices for a company:

- Day 1: 100
- Day 2: 105
- Day 3: 98
- Day 4: 102
- Day 5: 108

We want to calculate a 3-day moving average.



❖ **Calculation:**

1. Day 3: (Day 1 price + Day 2 price + Day 3 price) / 3 = (100 + 105 + 98) / 3 = 101
2. Day 4: (Day 2 price + Day 3 price + Day 4 price) / 3 = (105 + 98 + 102) / 3 = 101.67
3. Day 5: (Day 3 price + Day 4 price + Day 5 price) / 3 = (98 + 102 + 108) / 3 = 102.67

❖ **Result:** The 3-day moving averages for the given data are:

- Day 3: 101
- Day 4: 101.67
- Day 5: 102.67

❖ **Interpretation:** The moving average smooths out the fluctuations in the stock price. In this example, we can see that the moving average generally follows the upward trend of the stock price but with less volatility

❑ Discretization

- Many machine learning algorithms prefer or perform better when numerical input variables have a standard probability distribution, discrete form. Certain algorithms may be incompatible with continuous data.

- Continuous features have a smaller chance of correlating with the target variable. After discretizing a variable, groups corresponding to the target can be **interpreted better**.

- Discretization is the process through which we can transform continuous numerical variables into a discrete categorical variables. This involves dividing the range of a continuous variable into intervals (bins) and assigning data points to these bins based on their values.

- For example, we can divide a continuous variable, weight, and store it in the following groups : Under 100 gm (light), between 140–160 gm (mid), and over 200 gm (heavy). In the example, weights of *85* gm and *56* gm convey the same information (the object is light). Therefore, discretization helps make our data easier to understand if it fits the problem statement.

- Methods: Equal-width, Equal-frequency, clustered/ k-means

❑ **Equal-Width or Uniform Discretization :**

- Each bin has the same width in the span of possible values for the variable.
- Separating all possible values into ‘N’ number of bins, each having the same width. Formula for interval width:
  - Width = (maximum value - minimum value) / N, where N is the number of bins or intervals.
  - Example: 1-10: child; 11-20: teenager; 21-30: young; 31-40: middle-aged; 41-50: adults; 51-60: aged; 61-70: senior citizens;

❑ **Equal-Frequency or Quantile Discretization:**

- A quantile discretization transform will attempt to split the observations for each input variable into k groups, where the number of observations assigned to each group is approximately equal. For example, if we have 100 observations and we want 10 bins, each bin will have 10 observations.
- The advantage of this method is that it creates balanced bins that can handle skewed data and outliers better.

❑ **Clustered or K-Means Discretization:**

- Clusters are identified and observations / data points are assigned to each group.
- We apply K-Means clustering to the continuous variable, thus dividing it into discrete groups or clusters.

❑ Normalization

- Normalization is a technique used to transform the values of numerical features in a dataset to a common scale.
- Data normalisation transforms data in a way that they are either same scale and/or have similar distributions.
- This is essential because many machine learning algorithms are sensitive to the scale of the input features. If some features have a much larger range of values than others, they can dominate the learning process, leading to biased or inaccurate results.
- It offers several advantages, such as making data mining /machine learning algorithms more effective, faster data extraction, etc.
- This process of normalization is known by other names such as standardization, feature scaling etc.
- For example, consider a data set containing two features, age(x1), and income(x2), where age ranges from 20–80, while income ranges from 20,000-50,0000. Income is about several 1,000 times larger than age and ranges from 20,000–500,000. So, these two features are in very different ranges. When we do further analysis, like multivariate linear regression, for example, the attribute income will intrinsically influence the result more due to its larger value. But this doesn’t necessarily mean it is more important as a predictor.

- Methods:

- Rescaling: also known as “min-max normalization”; the values are shifted and rescaled so that they end up ranging between 0 and 1; it is the simplest of all methods and calculated as:

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

- Mean normalization: This method uses the mean of the observations in the transformation process:

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

- Z-score normalization: Also known as standardization, this technic uses Z-score or “standard score”. This method transforms the data so that it has a mean of 0 and a standard deviation of 1. It is widely used in machine learning algorithms such as SVM and logistic regression:

$$z = \frac{x - \mu}{\sigma}$$

Here, z is the standard score, ***μ*** is the population mean and ***σ*** is the population standard deviation

❑ Generalization

- Generalization is used to convert low-level data attributes to high-level data attributes by the use of concept hierarchy. It transforms one data value into a more imprecise one, a less specific value.
- Data generalization is the process of creating a broader categorization of data in a database, essentially ‘zooming out’ from the data to create a more general picture of trends or insights it provides.
- For examples: If you have a data set that includes the ages of a group of people, the data generalization process may look like this:
  - Original Data: Ages: 15, 17, 20, 26, 28, 31, 33, 37, 42, 42, 46, 48, 49, 54, 57, 57, 58, 59
  - Generalized Data: Ages: < 40 : young; >41: old
  - ❖ Here, an age in the numerical form of raw data (20, 52) is converted into (Young, old) categorical value.
- Customer Address Detail can be Customer City
- A date of birth could be generalized to become a month of birth or a year of birth.
- Data generalization replaces a specific data value with one that is less precise, which may seem counterproductive, but actually is a widely applicable and used technique in data mining, analysis, and secure storage.

THANK YOU

Any questions…?

JAIN DEEMED-TO-BE UNIVERSITY | SCHOOL OF COMPUTER SCIENCE AND IT

School of Computer Science & IT

Programme: BCA

INTRODUCTION TO DATA ANALYTICS (23BCAD4C01)

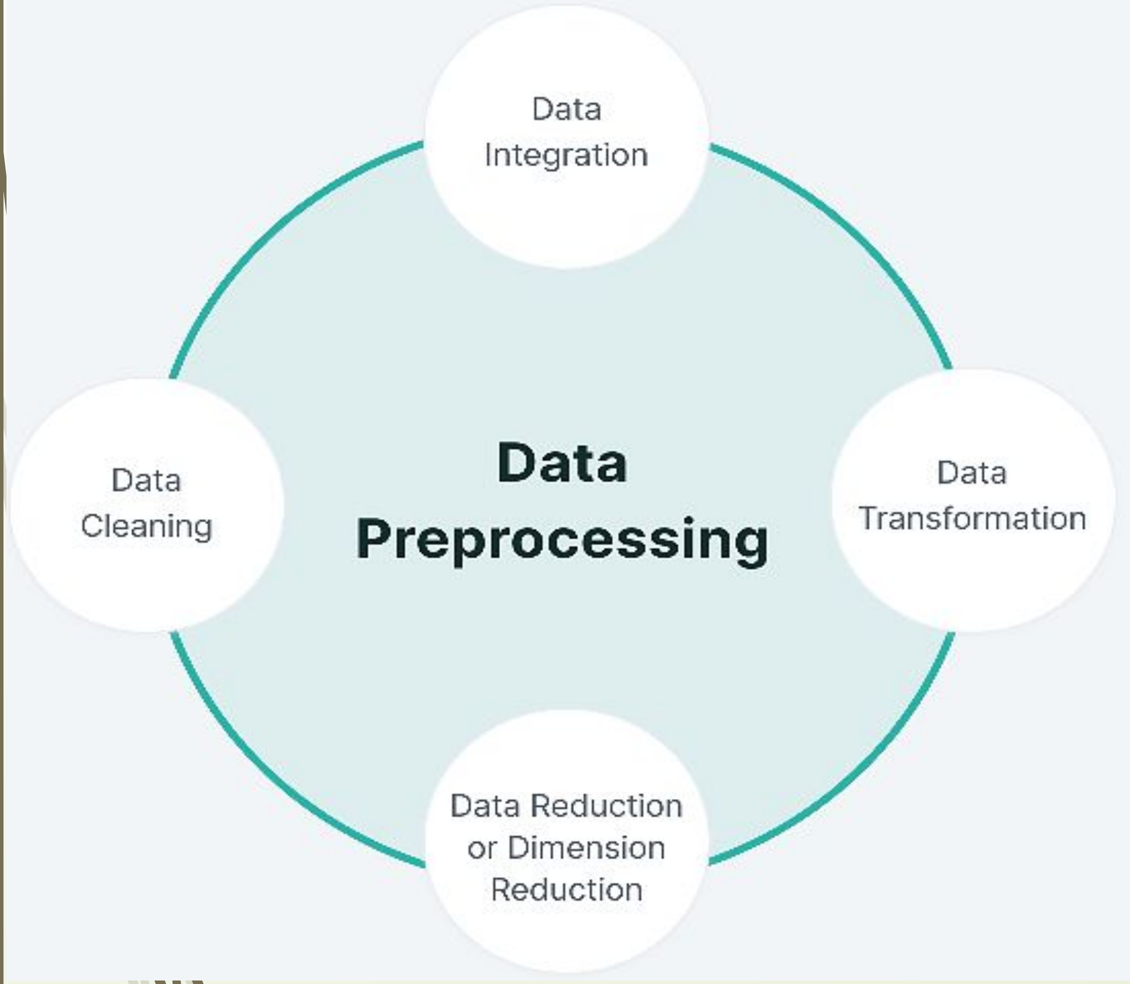
MODULE 2: Data Preparation

Dr. Ananta Charan Ojha, Professor

Session -4

- Data Pre-processing
  - Dimensionality Reduction

Major Tasks in Data Pre-processing



❑ Dimensionality Reduction

- Usually, datasets contain large number of attributes (i.e., features) and hundreds and thousands of instances (i.e., data points) making the dataset voluminous.
- High dimensional datasets throw several challenges in analytics projects.
- In particular, many machine learning algorithms can not be applied directly on a high dimensional dataset.
- Large amounts of data might sometimes produce worse performances in data analytics applications.
- Dimensionality reduction techniques help us in reducing the high dimensional data into a small dimensional data format that is easier to analyse, and visualize.

The Curse of Dimensionality

- The curse of dimensionality is a phenomenon that arises when you analyse and visualize with data in high-dimensional dataset.
- The higher is the number of features or variables in a dataset, the more difficult it becomes to visualize the training set and work on it.
- High dimensional data results complex models.
- Another vital point to consider is that most of the variables are often correlated.
- These correlated/ redundant variables within the feature set affect training of machine learning algorithms producing overfitted models that fail to perform well on real data.
- The primary aim of dimensionality reduction is to avoid overfitting. A training data with considerably lesser features will ensure that your model remains simple – it will make smaller assumptions.
- When you reduce the number of features and only keep the most relevant features, it is called **feature selection**.
- Therefore, the curse of dimensionality mandates the application of **dimensionality reduction**.

Benefits of Dimensionality Reduction

- Here are some of the benefits of applying dimensionality reduction to a dataset:

- It eliminates redundant features and noise making the hidden pattern more clearly visible. Helps in analysis and visualization of the data.
- It helps improve the accuracy and performance of the machine learning (ML) models.
- Some ML algorithms do not perform well when we have a large dimension. So, reducing these dimensions make the dataset compatible with the ML algorithm and helps usage of the algorithm.
- Space required to store the data is reduced as the number of dimensions comes down.
- Less dimensions lead to less computation/ and faster training time of the ML algorithm.

Techniques of Dimensionality Reduction

- There are several techniques for dimensionality reduction and feature selection. Commonly used techniques are outlined here.

1) Missing Value Ratio
2) Low Variance Filter
3) High Correlation Filter
4) Backward Feature Elimination
5) Forward Feature Selection
6) Random Forest
7) Principal Component Analysis

- **Missing Values Ratio**
  - Data attributes with too many missing values are unlikely to carry much useful information.
  - Thus data columns/ attributes with number of missing values greater than a given threshold (say 50%) can be removed.
  - The higher the threshold, the more aggressive the reduction.
- **Low Variance Filter**
  - Similarly to the previous technique, data columns with little changes in the data carry little information.
  - Thus all data columns with variance lower than a given threshold are removed.
  - A word of caution: variance is range dependent; therefore normalization is required before applying this technique.
- **High Correlation Filter**
  - High correlation between two variables means they have similar trends and are likely to carry similar information.
    - Suppose we have two variables: Income and Education. These variables will potentially have a high correlation as people with a higher education level tend to have significantly higher income, and vice versa.
  - In this case, only one of them will suffice to feed the machine learning model.
  - Here we calculate the correlation coefficient
    - Pearson's Product Moment Coefficient between numerical attributes and
    - Pearson's chi square value between nominal attributes as the case may be.
    - Pairs of columns with correlation coefficient higher than a threshold are reduced to only one.
  - A word of caution: correlation is scale sensitive; therefore attribute normalization is required for a meaningful correlation comparison.

Backward Feature Elimination

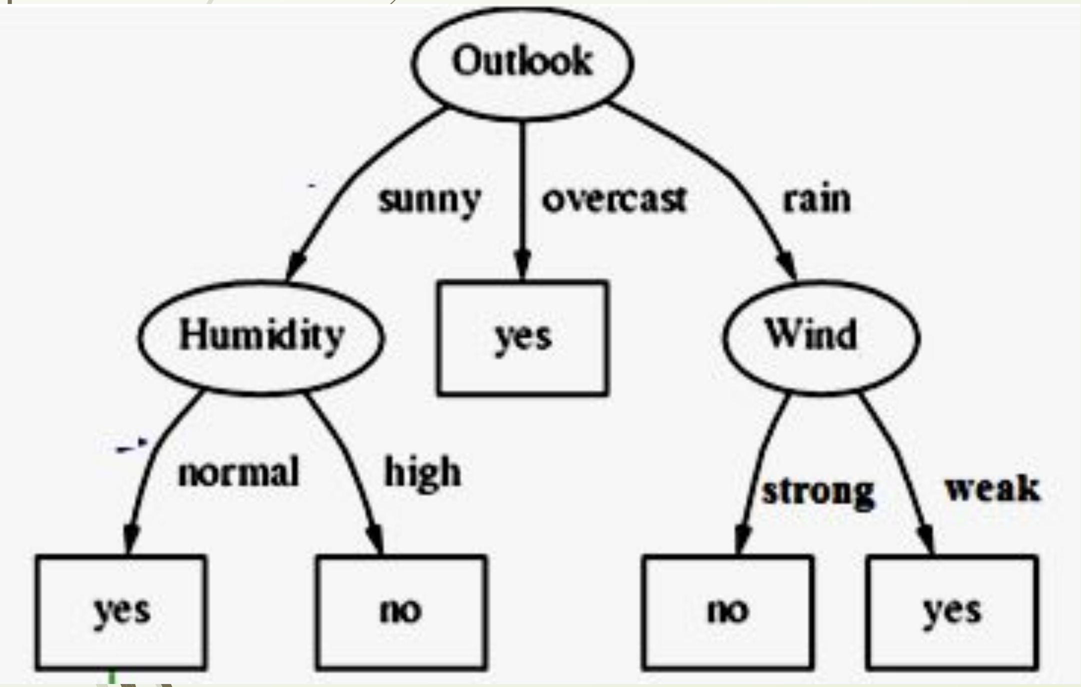
- In this technique,
  - We first take all the n attributes present in our dataset and train the model using them.
  - We then calculate the performance of the model (in terms of error, or prediction accuracy).
  - Now, we compute the performance of the model after eliminating each attribute (n times), i.e., we drop one attribute every time and train the model on the remaining n-1 attributes.
  - We identify the attribute whose removal has produced the smallest (or no) change in the performance of the model, and then drop that attribute (it means the attribute is not a good predictor, has no much effect on model performance).
  - Repeat this process until no attribute can be dropped.

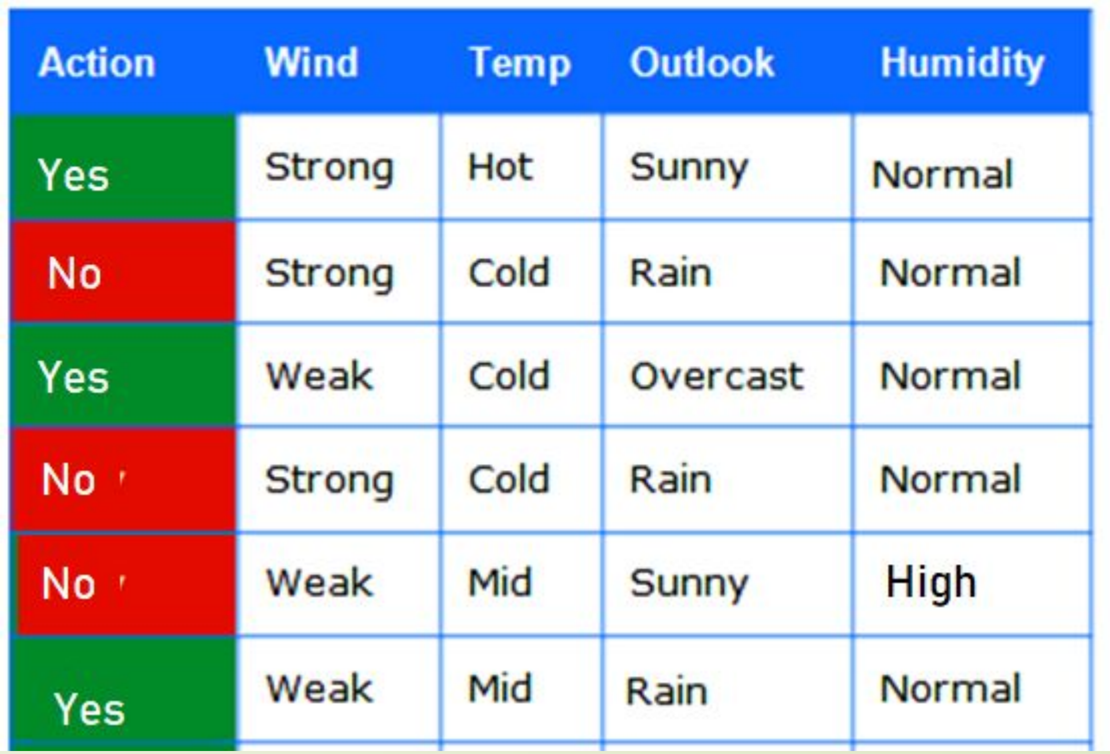
Forward Feature Selection

- This is the reverse process of the Backward Feature Elimination. Instead of eliminating features, we try to find the best features which improve the performance of the model.
- This technique works as follows:
  - We start with a single feature, progressively adding 1 feature at a time. Essentially, we train the model n number of times using each feature one after another.
  - The variable/attribute that produces the highest increase in performance is retained
  - We repeat this process until no significant improvement is seen in the model’s performance

Random Forest

- One approach to dimensionality reduction is to generate a large and carefully constructed set of trees against a target attribute and then use each attribute's usage statistics to find the most informative subset of features.
- Specifically, we can generate a large set (2000) of very shallow trees (2 levels), with each tree being trained on a small fraction (3) of the total number of attributes.
- If an attribute is often selected as best split, it is most likely an informative feature to retain.
- A score calculated on the attribute usage statistics in the random forest tells us – relative to the other attributes – which are the most predictive attributes.





Action	Wind	Temp	Outlook	Humidity
Yes	Strong	Hot	Sunny	Normal
No	Strong	Cold	Rain	Normal
Yes	Weak	Cold	Overcast	Normal
No	Strong	Cold	Rain	Normal
No	Weak	Mid	Sunny	High
Yes	Weak	Mid	Rain	Normal

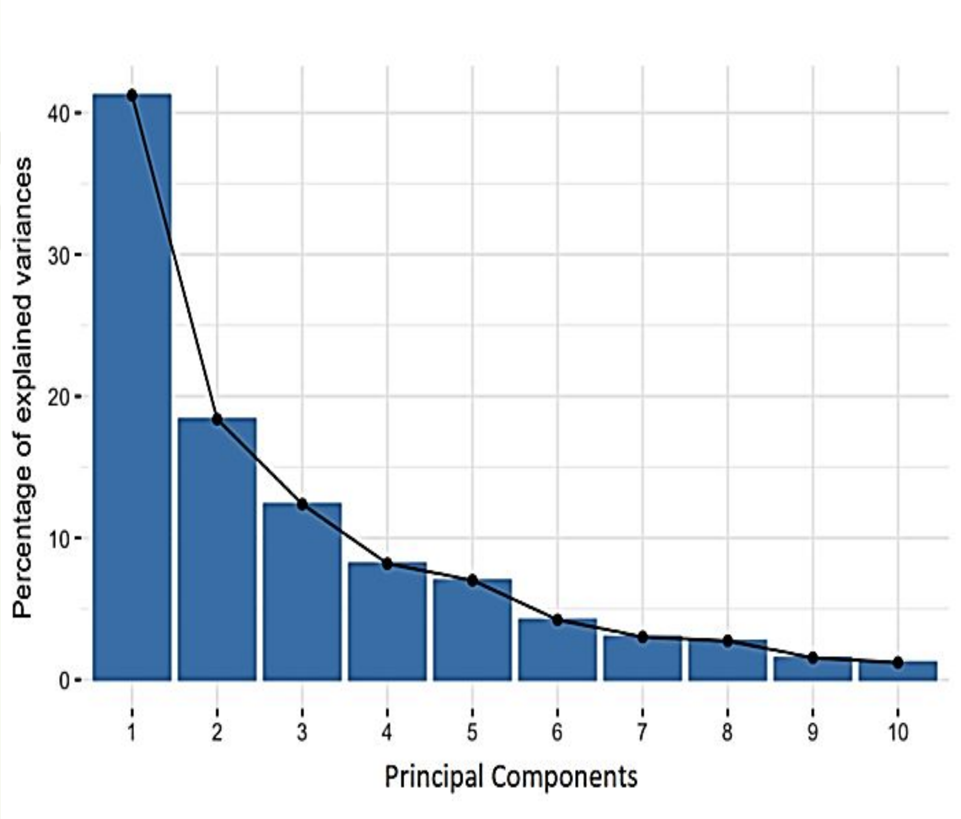
Example: Play Tennis

❑ Principal Component Analysis (PCA)

❖ PCA is a statistical technique which helps us in extracting a small set of transformed variables from a large set of original variables available in a dataset. The reduced set of transformed variables called Principal Component still contains most of the information in the large set. Smaller data sets are easier to explore and visualize and make data analysis process much easier and faster for machine learning algorithms.

- A principal component is a linear combination of the original variables:
- $Z^1 = \Phi^{11}X^1 + \Phi^{21}X^2 + \Phi^{31}X^3 + ... +\Phi^{p1}X^p$
- where,
  - Z^1 is first principal component
  - Φ^{p1} is the loading vector comprising of loadings (Φ^1, Φ^2..) of first principal component.

Eigenvectors, eigenvalues, and covariance matrix are computed in order to determine the principal components of the data.



✔ Principal components are extracted in such a way that the first principal component captures the maximum information / variance in the dataset. Larger the variability captured, larger the information captured by component.

✔ Second principal component is computed that tries to capture the remaining variance in the dataset and is uncorrelated (or orthogonal) to the first principal component.

✔ Third principal component is computed tries to capture the variance which is not captured by the first two principal components and so on.

✔ For example, if a 10-dimensional data gives you 10 principal components, PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on.

✔ Organizing information in principal components this way reduces dimensionality without losing much information, and helps us discarding the components with low information and considering the remaining components as important variables.

❖ The PCA conversion is sensitive to the variance of the relative scaling of the original variables. Thus, the data column ranges must first be normalized before implementing the PCA method.

THANK YOU

Any questions…?