Introduction To Data Science Final Syllabus

Lecture-1 : Introduction To Data Science

Q-1 : What is Big Data?

Big Data is any data that is expensive to manage and hard to extract value from.

Q-2 : Draw the figure of Big Data.

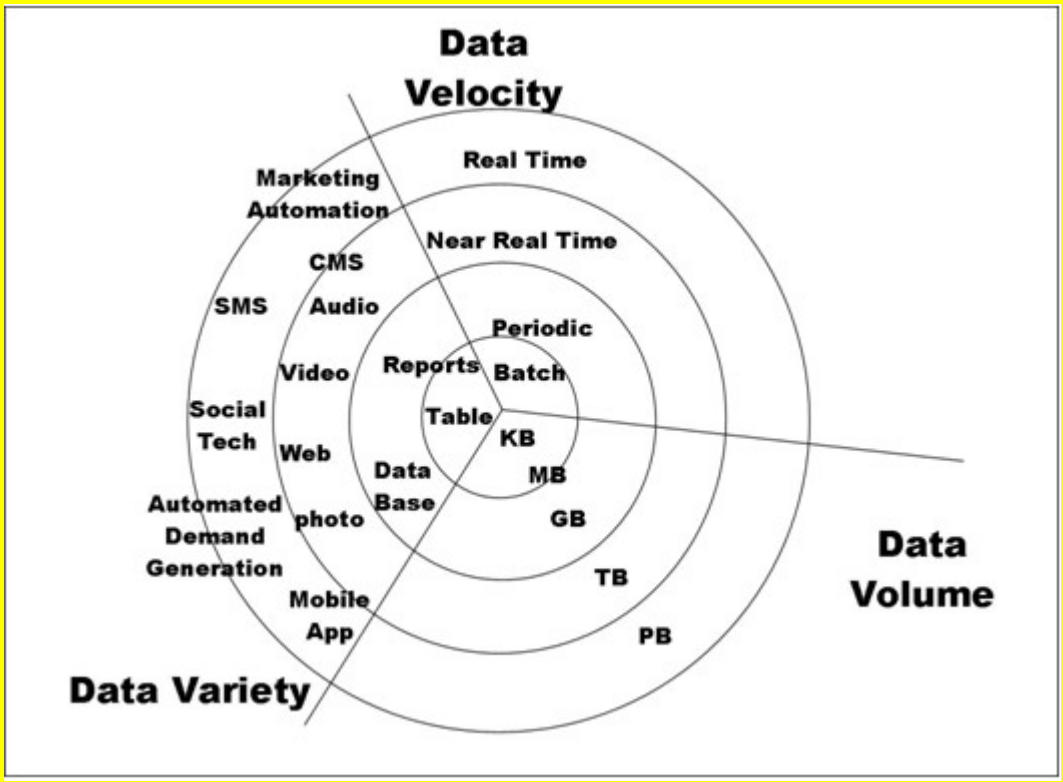

Fig-1 : Big Data

Q-3 : What are the characteristics of Big Data?

Big Data is characterized by three main dimensions. They are :

1. **Volume :** Volume refers the size of data.
2. **Velocity :** Velocity refers the latency of data processing relative to the growing demand for interactivity.
3. **Variety and Complexity :** Variety and Complexity means the diversity of sources, formats, quality and structures.

Q-4 : What are the relation of Volume, Velocity, Variety and Complexity?

Volume, velocity, variety, and complexity are all interconnected aspects of Big Data :

1. **Volume :** This refers to the sheer amount of data generated and collected. As the volume of data increases, so does the complexity, especially in terms of storage, processing, and analysis. Handling large volumes of data requires scalable infrastructure and efficient algorithms to manage and process it effectively.

2. **Velocity :** Velocity refers to the speed at which data is generated, collected, and processed. High velocity adds to the complexity of handling Big Data because it requires real-time or near-real-time processing capabilities. Rapidly changing data streams, such as social media feeds or sensor data, pose challenges in terms of capturing, processing, and analyzing data without delay.
3. **Variety and Complexity :** The variety of data, encompassing structured, semi-structured, and unstructured formats, adds a layer of complexity to Big Data analytics. Each type of data requires distinct processing methods and tools, necessitating integration from multiple sources while ensuring compatibility for analysis. As the volume and velocity of data increase, alongside its diverse nature, the complexity intensifies. Managing, processing, and analyzing large volumes of data in real-time or near-real-time demand sophisticated algorithms, distributed computing systems, and advanced analytics techniques to navigate the intricacies of Big Data effectively.

Q-5 : Describe the value chain with figure.

In data science, the term "value chain" refers to the sequence of activities or processes involved in extracting value from data. It encompasses the entire lifecycle of data—from its acquisition and storage to its analysis, interpretation, and application to derive meaningful insights and make informed decisions. The value chain in data science typically includes the following stages :

1. **Collection :** Getting the data.
2. **Engineering :** Storage and computational resources.
3. **Governance :** Overall management of data.
4. **Wrangling :** Data preprocessing and cleaning.
5. **Analysis :** Discovery (learning, visualization) etc.
6. **Presentation :** Arguing that results are significant and useful.
7. **Operationalization :** Putting the results to work.

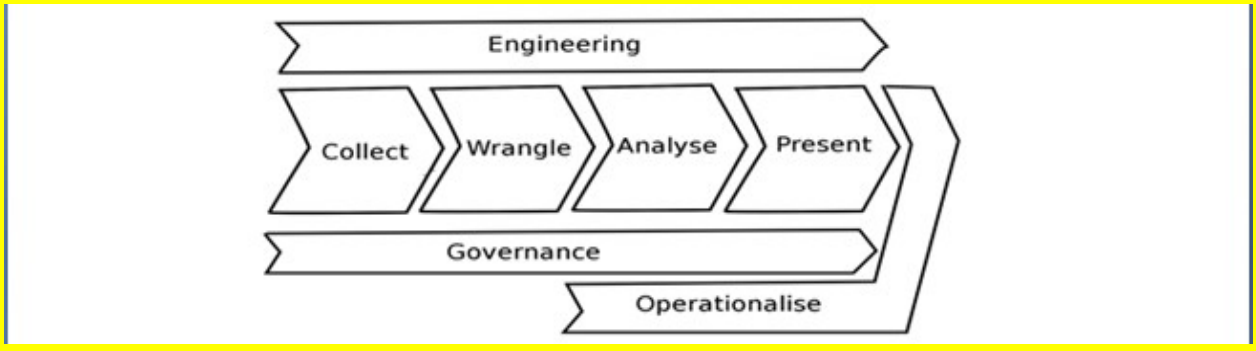

Fig-2 : The Value Chain

Q-6 : What are the activities of a data scientist?

Data scientists perform a wide range of activities across various stages of the data science lifecycle. These activities can vary depending on the organization, industry, and specific project requirements. However, some common activities of data scientists include :

1. **Data Collection :** Gathering data from multiple sources, including databases, files, APIs, web scraping, sensors, and social media platforms.
2. **Data Cleaning and Preprocessing :** Cleaning, filtering, and preprocessing raw data to ensure its quality, consistency, and suitability for analysis. This may involve handling missing values, removing duplicates, and transforming data into a usable format.
3. **Exploratory Data Analysis (EDA) :** Exploring and visualizing the data to understand its characteristics, distribution, patterns, and relationships. EDA helps data scientists gain insights and identify potential features for modeling.
4. **Feature Engineering :** Creating new features or transforming existing features to improve the performance of machine learning models. Feature engineering involves selecting, extracting, and encoding relevant information from the data.
5. **Model Development :** Building and training predictive or descriptive models using machine learning, statistical, or other analytical techniques. Data scientists select appropriate algorithms, tune model parameters, and evaluate model performance using validation techniques.
6. **Model Evaluation and Validation :** Assessing the performance of machine learning models using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Validation techniques such as cross-validation and holdout validation are used to ensure the generalization of models to unseen data.
7. **Model Deployment :** Deploying machine learning models into production environments, integrating them with existing systems, and making them accessible for real-time predictions or decision-making. This may involve collaboration with software engineers and DevOps teams.
8. **Monitoring and Maintenance :** Monitoring the performance of deployed models, detecting drift or degradation in model performance over time, and retraining models as needed to maintain their effectiveness.
9. **Collaboration and Communication :** Collaborating with cross-functional teams, including domain experts, business stakeholders, software engineers, and other data professionals. Communicating findings, insights, and recommendations effectively through reports, presentations, and visualizations.

10. Continuous Learning and Professional Development : Keeping abreast of the latest developments in data science, machine learning, and related fields. Engaging in continuous learning, attending conferences, workshops, and online courses to enhance skills and expertise.

These activities demonstrate the diverse and interdisciplinary nature of the data scientist's role, which requires proficiency in data manipulation, statistical analysis, machine learning, programming, and domain knowledge. Effective data scientists possess a combination of technical skills, critical thinking, and problem-solving abilities to extract meaningful insights from data and drive business value.

Q-7 : Define data science and data scientist from value chain.

From the perspective of the value chain, data science can be defined as the discipline focused on extracting value from data throughout its lifecycle. It encompasses a series of interconnected activities aimed at leveraging data to drive insights, make informed decisions, and create tangible outcomes for organizations. The value chain in data science represents the sequential stages involved in transforming raw data into actionable insights and value-added outcomes. These stages typically include data acquisition, preparation, analysis, interpretation, decision-making, and value creation.

A data scientist, within the context of the value chain, can be defined as a professional responsible for performing the activities across various stages of the data science lifecycle. Data scientists are skilled practitioners who possess expertise in data collection, cleaning, preprocessing, analysis, modeling, interpretation, and communication. They play a critical role in extracting insights from data, developing predictive or descriptive models, and translating analytical findings into actionable recommendations for decision-makers. Data scientists collaborate with cross-functional teams, including domain experts, business stakeholders, software engineers, and other data professionals, to drive data-driven decision-making and value creation within organizations.

In summary, data science and data scientists are integral components of the value chain, working together to unlock the potential of data and generate value for organizations through the application of analytical techniques and domain knowledge.

Lecture-2 : The Data Science Process

Q-1 : Why 50% of analytic projects fail? How to recover this?

There are several reasons why analytics projects fail, and addressing these challenges is essential to improving success rates. Some common reasons for failure include :

1. **Poorly Defined Objectives :** Lack of clarity or alignment on project objectives can lead to misinterpretation of results or irrelevant analyses.
2. **Data Quality Issues :** Inaccurate, incomplete, or inconsistent data can undermine the validity and reliability of analyses, leading to flawed conclusions.
3. **Lack of Stakeholder Involvement :** Insufficient engagement and collaboration with stakeholders can result in solutions that do not meet their needs or expectations.
4. **Insufficient Skills and Resources :** Inadequate expertise, tools, or resources can hinder the implementation of effective analytical solutions.
5. **Overly Complex Models :** Building overly complex models that are difficult to interpret or deploy can impede the usefulness and scalability of analytical solutions.
6. **Resistance to Change :** Organizational resistance to adopting data-driven approaches or implementing recommended changes can limit the impact of analytics projects.

To recover from these challenges and improve the success rate of analytics projects, organizations can take several steps :

1. **Define Clear Objectives :** Clearly define project objectives, scope, and success criteria in collaboration with stakeholders to ensure alignment and focus.
2. **Address Data Quality Issues :** Invest in data quality assessment, cleansing, and governance processes to ensure that data used for analysis is accurate, complete, and reliable.
3. **Engage Stakeholders :** Involve stakeholders throughout the project lifecycle to gather requirements, provide feedback, and ensure that analytical solutions meet their needs and expectations.
4. **Develop Skills and Resources :** Invest in training, hiring, or partnering with skilled data professionals and providing them with the necessary tools and resources to effectively execute analytics projects.
5. **Simplify Models :** Focus on building simpler, interpretable models that strike a balance between accuracy and complexity, making them easier to understand, deploy, and maintain.
6. **Promote Change Management :** Implement change management strategies to address organizational culture, communication, and leadership issues that may impede the adoption of data-driven approaches.
7. **Iterative Approach :** Adopt an iterative and agile approach to analytics projects, allowing for continuous feedback, experimentation, and improvement throughout the project lifecycle.

By addressing these challenges and implementing these strategies, organizations can increase the likelihood of success for analytics projects and realize the full potential of data-driven decision-making.

Q-2 : What is Data Preparation? Define 4 steps of Data Preparation.

Data preparation is the process of cleaning, transforming, and organizing raw data into a format suitable for analysis. It involves several steps to ensure that the data is accurate, complete, and structured in a way that facilitates effective analysis. One common framework for data preparation involves four key steps: Identify, Collect, Assess, and Vectorize.

1. **Identify :** In this initial step, the focus is on understanding the data requirements and identifying relevant data sources. This includes defining the scope of the analysis, determining the types of data needed, and identifying potential sources where the data can be found. It also involves identifying any constraints or limitations associated with the data sources, such as data availability, quality, or accessibility.
2. **Collect :** Once the data sources have been identified, the next step is to collect the necessary data. This may involve extracting data from databases, files, APIs, web scraping, sensors, or other sources. Data collection methods should be chosen based on the availability and accessibility of the data sources. It's important to ensure that the collected data is comprehensive and covers the relevant variables needed for analysis.
3. **Assess :** After collecting the data, it's essential to assess its quality, consistency, and completeness. This involves performing data profiling and exploratory data analysis (EDA) to understand the characteristics of the data, identify any anomalies or outliers, and detect any data quality issues, such as missing values, duplicates, or errors. Data assessment helps to determine the suitability of the data for analysis and informs decisions about data cleaning and preprocessing.
4. **Vectorize :** The final step in data preparation is to transform the data into a structured format suitable for analysis. This involves converting categorical variables into numerical representations (e.g., one-hot encoding), scaling numerical variables to a common scale, and handling missing values through imputation or deletion. Vectorization also includes feature engineering, where new features are created or existing features are transformed to enhance the predictive power of the data. The goal of vectorization is to prepare the data in a format that can be fed into machine learning algorithms for analysis.

By following these four steps of data preparation—Identify, Collect, Assess, and Vectorize—data scientists can ensure that the data is clean, consistent, and well-structured, laying the foundation for effective analysis and decision-making.

Q-3 : What is Data Aggregation? What is the role of Data Aggregation?

Data aggregation is the process of combining and summarizing data from multiple sources or granular levels into a more concise and informative representation. This process typically involves grouping data based on certain criteria and then applying an aggregation function (such as sum, average, count, min, max, etc.) to the grouped data to generate aggregated results. Data aggregation is commonly used in data analysis and reporting to simplify complex datasets, reduce redundancy, and extract meaningful insights.

The role of data aggregation is to condense large volumes of data into a more manageable and understandable format, making it easier to analyze and interpret trends, patterns, and relationships within the data. It helps in gaining a higher-level perspective and understanding of the underlying data, which can inform decision-making and strategic planning.

For example, consider a retail company that collects transaction data from its stores. The raw transaction data may consist of individual sales transactions recorded for each customer, including details such as the date, time, store location, product purchased, quantity, and price. Analyzing this raw transaction data at the individual transaction level may be overwhelming and inefficient for identifying overall sales trends or performance.

To address this, the retail company may aggregate the transaction data by store location and date, calculating metrics such as total sales revenue, average transaction value, and total number of transactions for each store and day. By aggregating the data in this way, the company can gain insights into sales performance across different stores and over time, identify peak sales periods, and compare performance between stores. This aggregated data can then be used for strategic decision-making, such as optimizing inventory management, staffing levels, or marketing campaigns.

In summary, data aggregation plays a crucial role in simplifying and summarizing complex datasets, enabling organizations to extract meaningful insights and make informed decisions based on aggregated results.

Q-4 : What is Feature Reduction? Why we reduce feature?

Feature reduction, also known as dimensionality reduction, is the process of reducing the number of input variables or features in a dataset while preserving the most

relevant information. This technique is commonly used in data preprocessing and machine learning to address the curse of dimensionality, where datasets with a large number of features can lead to increased computational complexity, overfitting, and decreased model performance.

There are several reasons why feature reduction is performed :

1. **Simplicity :** Simplifying the dataset by reducing the number of features can make it easier to understand, interpret, and analyze. It can also facilitate visualization of high-dimensional data.
2. **Computational Efficiency :** By reducing the dimensionality of the dataset, the computational resources required for processing and analyzing the data decrease, leading to faster training and inference times for machine learning models.
3. **Overfitting Prevention :** High-dimensional datasets are prone to overfitting, where a model learns to capture noise or irrelevant patterns in the data, leading to poor generalization performance on unseen data. Feature reduction helps mitigate overfitting by focusing on the most informative features and reducing the risk of model complexity.
4. **Improved Generalization :** By focusing on the most relevant features, feature reduction can improve the generalization performance of machine learning models by reducing the impact of noise and irrelevant information in the data.
5. **Addressing Multicollinearity :** Feature reduction can help address multicollinearity, where features are highly correlated with each other, by selecting a subset of features that capture the essential information without redundancy.

There are various techniques for feature reduction, including :

- **Feature Selection :** Selecting a subset of the original features based on their relevance or importance to the target variable. Common methods include filter methods (e.g., correlation, mutual information), wrapper methods (e.g., recursive feature elimination), and embedded methods (e.g., feature importance from tree-based models).
- **Feature Extraction :** Transforming the original features into a lower-dimensional space using techniques such as principal component analysis (PCA), linear discriminant analysis (LDA), or t-distributed stochastic neighbor embedding (t-SNE). These methods project the data onto a new set of orthogonal or linearly independent features that capture the most significant variability in the data.

By performing feature reduction, data scientists can improve the efficiency, interpretability, and generalization performance of machine learning models, leading to better decision-making and insights from the data.

Q-5 : Define Testing in Model Training? How Testing is performed? Why Testing is needed? What are the Types of Testing used in Model Training?

In model training, testing refers to the process of evaluating the performance and generalization ability of a trained machine learning model on unseen or held-out data. Testing is a critical step in the model development lifecycle and is essential for assessing how well the model can make predictions or classifications on new, unseen examples. Testing helps to measure the model's accuracy, reliability, and robustness, allowing data scientists to identify potential issues, refine the model, and make informed decisions about its deployment and use in real-world scenarios.

Testing is performed by using a separate dataset, known as the test set, that is distinct from the data used for training the model. The test set contains examples that the model has not seen during training, ensuring an unbiased evaluation of its performance on unseen data. The model is applied to the test set to make predictions or classifications, and its performance is evaluated using various metrics, such as accuracy, precision, recall, F1-score, or area under the receiver operating characteristic curve (ROC-AUC).

Testing is needed for several reasons :

1. **Assess Model Performance :** Testing helps to assess how well the trained model performs on unseen data, providing insights into its accuracy and generalization ability.
2. **Detect Overfitting :** Testing helps to detect overfitting, where a model learns to capture noise or irrelevant patterns in the training data, leading to poor performance on unseen data.
3. **Optimize Hyperparameters :** Testing allows data scientists to tune model hyperparameters (e.g., learning rate, regularization strength) based on their impact on test set performance, improving the model's performance and generalization ability.
4. **Compare Models :** Testing enables the comparison of different models or algorithms to determine which one performs best on the test set, helping to guide model selection and refinement.

5. **Validate Model Assumptions :** Testing helps to validate assumptions made during model development and ensure that the model behaves as expected in real-world scenarios.

There are several types of testing used in model training including :

1. **A/B Testing :** A/B testing, also known as split testing, involves comparing the performance of two or more variants of a model (or different models) by randomly assigning users or data samples to different groups and measuring their responses. This approach is commonly used in web applications, marketing campaigns, and product development to assess the impact of changes or interventions.
2. **Fractional Testing :** Fractional testing involves randomly partitioning the dataset into training, validation, and test sets using a predefined fraction of the data. Typically, a larger portion of the data is used for training, while smaller fractions are allocated for validation and testing. This approach helps ensure that the model's performance is evaluated on diverse subsets of the data.
3. **Factorial Testing :** Factorial testing, also known as factorial design, involves systematically varying multiple factors or parameters of the model simultaneously to assess their combined effects on performance. This approach allows us to explore the interactions between different factors and optimize the model's performance across multiple dimensions.

Each type of testing has its advantages and is suitable for different use cases and scenarios. By performing testing during model training, data scientists can validate the model's performance, detect potential issues, and iteratively improve its accuracy and reliability before deployment in real-world applications.

Lecture-3 : Statistical Inference

Q-1 : What is Point Estimate? What is the relation of Sampling Distribution and Point Estimate? Which Sampling Distribution is better? Why?

A point estimate is a single value that is used to estimate or infer a population parameter based on sample data. It provides an approximation of the true value of the parameter, which is often unknown or impractical to measure for the entire population. Point estimates are commonly used in statistical inference to make predictions, draw conclusions, or test hypotheses about the population.

For example, if we want to estimate the mean height of all adults in a country, we might take a random sample of individuals and calculate the mean height of that sample. The calculated mean height from the sample would serve as a point estimate of the population mean height.

The relationship between sampling distribution and point estimate is as follows :

- **Sampling Distribution :** The sampling distribution is the distribution of a statistic (e.g., mean, proportion) calculated from multiple samples taken from the same population. It represents the variability of the statistic across different samples and provides information about the sampling error, which is the discrepancy between the sample statistic and the population parameter.
- **Point Estimate :** A point estimate is a single value derived from a sample statistic that serves as an estimate of a population parameter. It is obtained from one particular sample and represents our best guess of the true population parameter based on that sample.

The sampling distribution plays a crucial role in determining the reliability and precision of a point estimate. Ideally, we want a sampling distribution that is centered around the true population parameter and has low variability (i.e., low standard error), indicating that the point estimates are close to the true value and consistent across different samples.

In terms of which sampling distribution is better, it depends on the specific context and requirements of the analysis :

- **Large Sample Size :** When the sample size is large, the sampling distribution tends to approximate a normal distribution due to the central limit theorem. In this case, point estimates such as the sample mean or proportion are more likely to be accurate and reliable, making the sampling distribution better for making inferences about the population.
- **Small Sample Size :** When the sample size is small, the sampling distribution may deviate from a normal distribution, and the variability of point estimates may be higher. In such cases, alternative methods or robust estimators may be preferred to obtain more accurate point estimates and account for the limitations of small sample sizes.

In summary, the quality of a point estimate depends on the characteristics of the sampling distribution, including its shape, center, and variability. A sampling distribution with low variability and centered around the true population parameter is desirable for obtaining accurate and reliable point estimates.

Q-2 : What is Hypothesis Testing? Convert the following word hypotheses into statistical hypotheses :

1. ***People who eat breakfast will run a race faster or slower than those who do not eat breakfast.***
2. ***People who own cats will live longer than those who do not own cats.***
3. ***People who earn an A in statistics are more likely to be admitted to graduate school than those who do not earn an A.***

Hypothesis testing is a statistical method used to make inferences about population parameters based on sample data. It involves formulating two competing hypotheses, the null hypothesis (H0) and the alternative hypothesis (H1 or Ha), and conducting statistical tests to determine whether there is enough evidence to reject the null hypothesis in favor of the alternative hypothesis.

The null hypothesis (H_0) typically represents the status quo or the absence of an effect, while the alternative hypothesis (H_1 or H_a) represents the hypothesis of interest or the presence of an effect.

Now, let's convert the word hypotheses into statistical hypotheses :

1. **Word Hypothesis :** People who eat breakfast will run a race faster or slower than those who do not eat breakfast. Statistical Hypotheses :
   - **Null Hypothesis ($\mathbf{H_0}$) :** The mean race time for people who eat breakfast is equal to the mean race time for people who do not eat breakfast. $H_0 : \mu_{\_breakfast} = \mu_{\_no\_breakfast}$.
   - **Alternative Hypothesis ($\mathbf{H_1}$) :** The mean race time for people who eat breakfast is different from the mean race time for people who do not eat breakfast. $H_1 : \mu_{\_breakfast} \neq \mu_{\_no\_breakfast}$.
2. **Word Hypothesis :** People who own cats will live longer than those who do not own cats. Statistical Hypotheses :
   - **Null Hypothesis ($\mathbf{H_0}$) :** The mean lifespan for people who own cats is equal to the mean lifespan for people who do not own cats. $H_0 : \mu_{\_cats} = \mu_{\_no\_cats}$.
   - **Alternative Hypothesis ($\mathbf{H_1}$) :** The mean lifespan for people who own cats is greater than the mean lifespan for people who do not own cats. $H_1 : \mu_{\_cats} > \mu_{\_no\_cats}$.
3. **Word Hypothesis :** People who earn an A in statistics are more likely to be admitted to graduate school than those who do not earn an A. Statistical Hypotheses :
   - **Null Hypothesis ($\mathbf{H_0}$) :** The proportion of students admitted to graduate school is the same for those who earn an A in statistics and those who do not earn an A. $H_0 : p_A = p_{\_no\_A}$.

- **Alternative Hypothesis $(\mathbf{H_1})$:** The proportion of students admitted to graduate school is higher for those who earn an A in statistics compared to those who do not earn an A. H_1 : $\mathrm{p\_A} > p\_no\_A$.

In hypothesis testing, we collect sample data and use statistical methods to determine whether the evidence supports rejecting the null hypothesis in favor of the alternative hypothesis.

Lecture-4 : Simple Linear Regression

Q-1 : What is Residual Plots? What are the Residual Plots about? What are the application of Residual Plot for Data Scientist?

Residual plots are graphical tools used in regression analysis to assess the goodness of fit of a regression model and to diagnose potential problems or violations of model assumptions. They are plots of the residuals, which are the differences between the observed values and the predicted values from the regression model, against the predictor variables or fitted values.

The main purposes of residual plots are :

1. **Assessing Model Assumptions :** Residual plots help data scientists check whether the assumptions of the regression model are met. These assumptions include linearity, constant variance of errors (homoscedasticity), normality of errors, and independence of errors.
2. **Detecting Patterns or Trends :** Residual plots can reveal patterns or trends in the residuals, indicating potential violations of the assumptions or problems with the model. Common patterns include nonlinearity, heteroscedasticity, and outliers.
3. **Identifying Influential Observations :** Residual plots can identify influential observations that have a disproportionate impact on the regression model's parameters or predictions. Outliers or leverage points may be evident in the residual plot as points with large residuals.
4. **Model Improvement :** Residual plots can guide model improvement by suggesting modifications to the model structure, such as adding polynomial terms, transforming variables, or considering alternative regression techniques.
5. **Validiting Predictive Performance :** Residual plots can be used to assess the predictive performance of the regression model. A well-fitted model should have residuals that are randomly scattered around zero without any discernible patterns.

The application of residual plots for data scientists includes :

- **Model Diagnosis :** Data scientists use residual plots to diagnose potential issues with regression models and assess their validity. By examining the patterns or trends in the residuals, data scientists can identify areas for model improvement or refinement.
- **Model Selection :** Residual plots can aid data scientists in selecting the most appropriate regression model among competing models. Comparing residual plots for different models helps identify the model that best fits the data and meets the assumptions of regression analysis.
- **Model Interpretation :** Residual plots provide insights into the relationship between the predictor variables and the response variable. Data scientists use residual plots to interpret the effects of predictor variables on the response and assess the overall goodness of fit of the model.
- **Quality Assurance :** Residual plots serve as a quality assurance tool for regression analysis, helping data scientists ensure the reliability and validity of their models before making predictions or drawing conclusions based on the results.

Overall, residual plots are valuable tools for data scientists in regression analysis, providing visual diagnostics and guiding model development, interpretation, and validation.

Q-2 : Example on Weekly Advertising Expenditure (Find out Standard Error of the mean, T Distribution Table will be given).

Example: weekly advertising expenditure

y	x	y-hat	Residual (e)
1250	41	1270.8	-20.8
1380	54	1411.2	-31.2
1425	63	1508.4	-83.4
1425	54	1411.2	13.8
1450	48	1346.4	103.6
1300	46	1324.8	-24.8
1400	62	1497.6	-97.6
1510	61	1486.8	23.2
1575	64	1519.2	55.8
1650	71	1594.8	55.2

$\hat{y} = 828 + 10.8x$ and $e_i = y_i - \hat{y}_i$

20

Fig-3 : Example on weekly advertising expenditure (part-1)

Regression Standard Error

y	x	y-hat	Residual (e)	square(e)
1250	41	1270.8	-20.8	432.64
1380	54	1411.2	-31.2	973.44
1425	63	1508.4	-83.4	6955.56
1425	54	1411.2	13.8	190.44
1450	48	1346.4	103.6	10732.96
1300	46	1324.8	-24.8	615.04
1400	62	1497.6	-97.6	9525.76
1510	61	1486.8	23.2	538.24
1575	64	1519.2	55.8	3113.64
1650	71	1594.8	55.2	3047.04
y-hat = 828+10.8X			total	36124.76
			$S_{y.x}$	67.19818

23

Fig-4 : Example on weekly advertising expenditure (part-2)

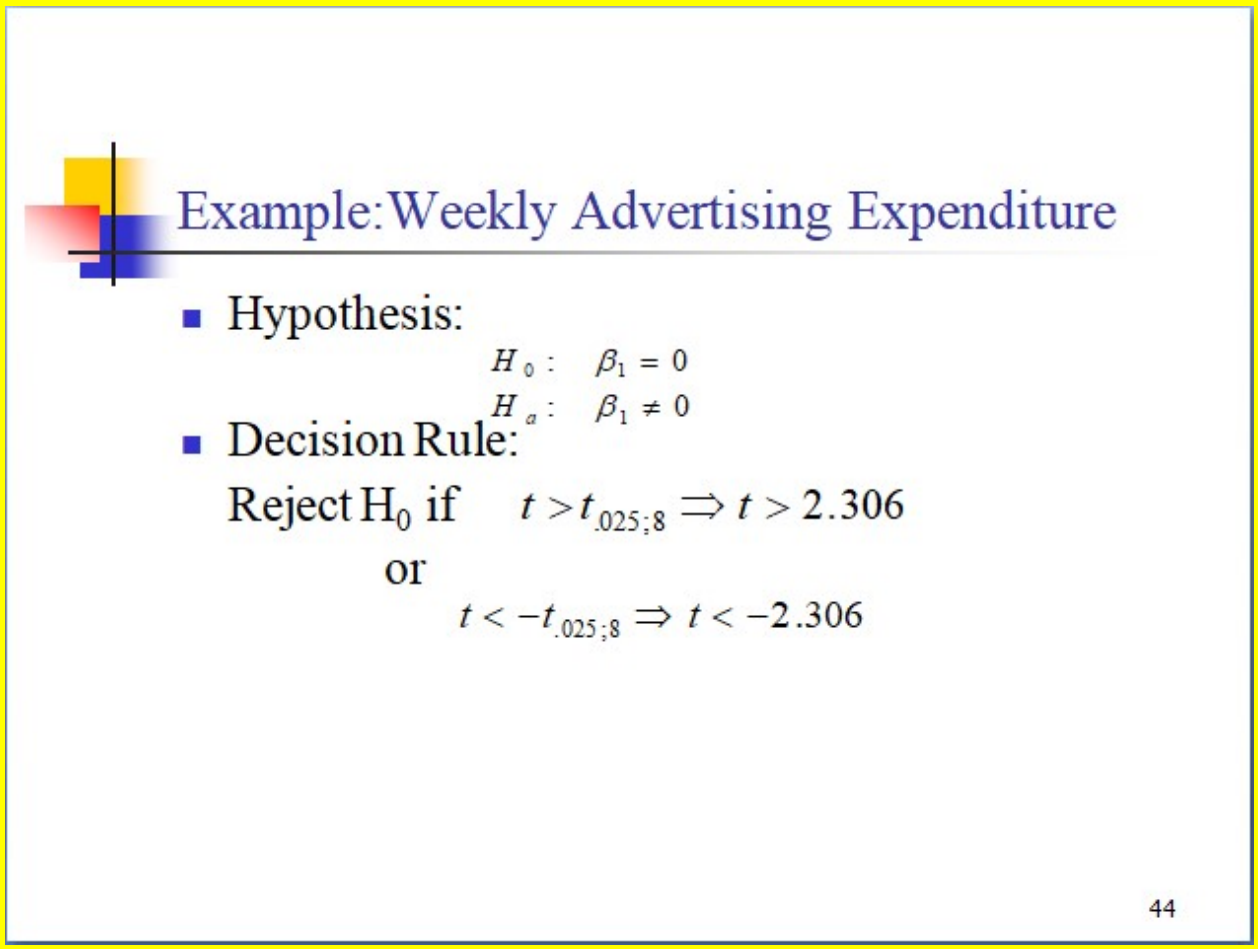

Fig-5 : Example on weekly advertising expenditure (part-3)

Lecture-5 : Logistic Regression

Q-1 : Define maximum likelihood with example.

Maximum likelihood estimation (MLE) is a statistical method used to estimate the parameters of a probability distribution that best explain the observed data. The principle behind maximum likelihood estimation is to find the set of parameter values that maximizes the likelihood function, which measures the probability of observing the given data under the assumed probability distribution.

Here's a step-by-step explanation of maximum likelihood estimation :

1. **Assume a Probability Distribution :** First, we need to make an assumption about the probability distribution that describes the data. The choice of distribution depends on the nature of the data and the problem at hand. Common distributions include the normal distribution, binomial distribution, Poisson distribution, exponential distribution, etc.
2. **Formulate the Likelihood Function :** The likelihood function $L(\theta|x_1, x_2, \ldots, x_n)$ represents the probability of observing the given data $x_1, x_2, \ldots, x_n$ given a set of

parameter values θ of the assumed distribution. It is calculated as the joint probability density (or mass) function evaluated at the observed data points.

3. **Maximize the Likelihood Function :** The goal of maximum likelihood estimation is to find the values of the parameters θ that maximize the likelihood function. This is typically done using optimization techniques such as gradient descent, Newton-Raphson method, or numerical optimization algorithms.
4. **Estimate the Parameters :** Once the likelihood function is maximized, the estimated values of the parameters $\hat{\theta}$ are obtained. These estimated parameters represent the maximum likelihood estimates and are used as the best estimates of the true parameter values given the observed data.
5. **Assess the Model Fit :** Finally, the goodness of fit of the model can be assessed by evaluating the likelihood function at the maximum likelihood estimates and comparing it to alternative models or hypothesis testing.

Here's an example to illustrate maximum likelihood estimation :

Suppose we have a sample of n independent and identically distributed observations from a normal distribution with unknown mean μ and known standard deviation σ. Our goal is to estimate the parameter μ using maximum likelihood estimation.

1. **Assume a Probability Distribution :** We assume that the data follows a normal distribution $N(\mu, \sigma^2)$.
2. **Formulate the Likelihood Function :** The likelihood function for a sample of n observations $x_1, x_2, \ldots, x_n$ is given by :

   $$L(\mu|x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$$
3. **Maximize the Likelihood Function :** We maximize the likelihood function with respect to μ to find the value that maximizes the probability of observing the given data.
4. **Estimate the Parameter :** The estimated value of μ, denoted as $\hat{\mu}$, is obtained by maximizing the likelihood function.
5. **Assess the Model Fit :** The goodness of fit of the normal distribution model can be assessed by evaluating the likelihood function at the estimated value of μ and comparing it to alternative models or hypothesis testing.

In summary, maximum likelihood estimation is a powerful method for estimating the parameters of a probability distribution based on observed data, and it is widely used in various fields such as statistics, machine learning, and econometrics.

Q-2 : Prove that, $ln\,L = \sum_i y_i \beta x_i - \sum_i ln\left(1 + exp(\beta x_i)\right)$. (Fill in the gap there. Gap should be verified. The given equation should be proved with the equation $ln\left(\frac{P(Y)}{1-P(Y)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_K X_K = \beta x_i$.

To prove the given equation $\ln L = \sum_i y_i \beta x_i - \sum_i \ln\left(1 + \exp(\beta x_i)\right)$, we start with the logistic regression model :

$$\ln\left(\frac{P(Y)}{1-P(Y)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_K X_K = \beta x_i$$

where $P(Y)$ is the probability of the dependent variable $Y, X_1, X_2, \ldots, X_K$ are the independent variables, $\beta_0, \beta_1, \ldots, \beta_K$ are the regression coefficients, and x_i is the linear combination of the independent variables.

Taking the exponential of both sides, we get :

$$\frac{P(Y)}{1-P(Y)} = \exp(\beta x_i)$$

Solving for $P(Y)$, we have :

$$P(Y) = \frac{\exp(\beta x_i)}{1+\exp(\beta x_i)}$$

Now, the likelihood function L is the joint probability of observing the dependent variable Y given the independent variables $X_1, X_2, \ldots, X_K$. Assuming the observations are independent and identically distributed, we can write the likelihood function as the product of the probabilities of observing each individual outcome y_i :

$$L = \prod_i P(Y = y_i)$$

Substituting the expression for $P(Y)$, we have :

$$L = \prod_i \left(\frac{\exp(\beta x_i)}{1+\exp(\beta x_i)}\right)^{y_i} \left(\frac{1}{1+\exp(\beta x_i)}\right)^{1-y_i}$$

Taking the natural logarithm of both sides, we get :

$$\ln L = \sum_i y_i \ln\left(\frac{\exp(\beta x_i)}{1+\exp(\beta x_i)}\right) + (1 - y_i) \ln\left(\frac{1}{1+\exp(\beta x_i)}\right)$$

$$= \sum_i y_i \ln\left(\exp(\beta x_i)\right) - \sum_i \ln \ln\left(1 + \exp(\beta x_i)\right)$$

$$= \sum_i y_i \beta x_i - \sum_i \ln\left(1 + \exp(\beta x_i)\right)$$

Thus, we have proved that $\ln L = \sum_i y_i \beta x_i - \sum_i \ln\left(1 + \exp(\beta x_i)\right)$, as required.

Q-3 : Problem on ART (Assisted Reproduction Technology) with the equation $ln\left(\frac{Pr(pregnancy)}{1-Pr(pregnancy)}\right) = 2.67 - 0.13 * Age$ or $Age = exp(-0.13) = 0.88$ or $Pr(pregnancy) = \frac{exp(2.67-0.13*Age)}{1+exp(2.67-0.13*Age)}$.

Q1. What is the effect of Age on Pregnancy?

A. The $\hat{OR}_{Age} = \exp(-0.13) = 0.88$

This implies that for every 1 yr. increase in age, the odds of pregnancy decrease by 12%.

Fig-6 : Problem on ART (Problem-1)

Q2. What is the predicted probability of a 25 yr. old having pregnancy success with first ART attempt?

$$\hat{\Pr}(\text{pregnancy}) = \frac{\exp(2.67 - 0.13 * 25)}{1 + \exp(2.67 - 0.13 * 25)} = 0.359$$

A. From this model, a 25 yr. old has about a 36% chance of pregnancy success.

Fig-7 : Problem on ART (Problem-2)

Lecture-6 : Machine Learning

Q-1 : What are the requirements of Nearest-Neighbor Classifiers? Provide algorithm of Nearest Neighbor Classifier.

Nearest Neighbor Classifier requires three things. They are :

1. The set of stored records.
2. Distance Metric to compute distance between records.
3. The value of *k*, the number of nearest neighbors to retrieve.

Algorithm of Nearest Neighbor Classifier to classify an unknown record is given below.

Step-1 : Compute distance to other training records.

Step-2 : Identify k nearest neighbors.

Step-3 : Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote).

Q-2 : Define the problem of Nearest Neighbor Classifier in choosing the value of k for both large and small value. / What type of problem can be

arise in the given figure for the value of k in Nearest Neighbor Classification Method? (if figure is given).

Problem of Nearest Neighbor Classifier in choosing the value of k :

- If k is too small, sensitive to noise point.
- If k is too large, neighborhood may include points from other classes.

Q-3 : Why KNN classifiers are said to be lazy learners? How to solve it?

KNN classifier are lazy learners because :

- It does not build models explicitly.
- Unlike eager learners such as decision tree induction and rule-based systems.

To solve the problem of KNN classifier i.e. to reduce its laziness we can use the kd-tree algorithm. The algorithm works according to the following steps :

Step-1 : Select the x or y dimension (alternating between the two).

Step-2 : Partition the space into two with a line passing from the median point.

Step-3 : Repeat recursively in the two partitions as long as there are enough points.

Q-4 : Problem of kd-tree algorithm.

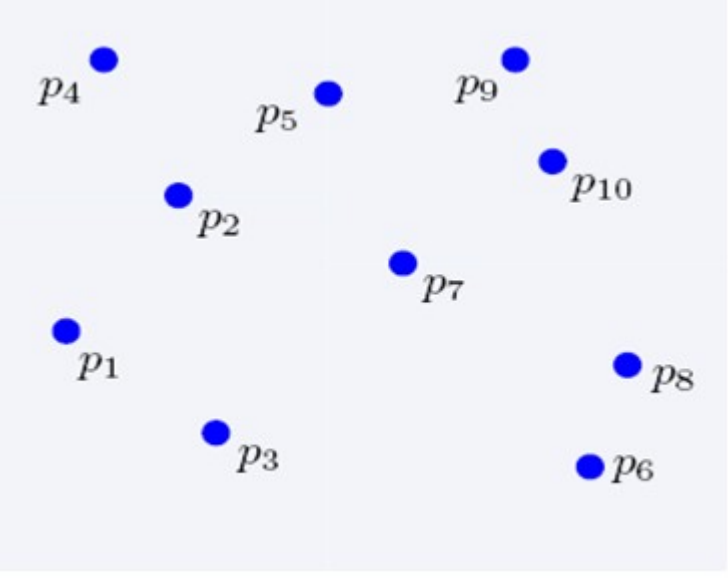

Fig-8 : Constructing kd-tree (step-1)

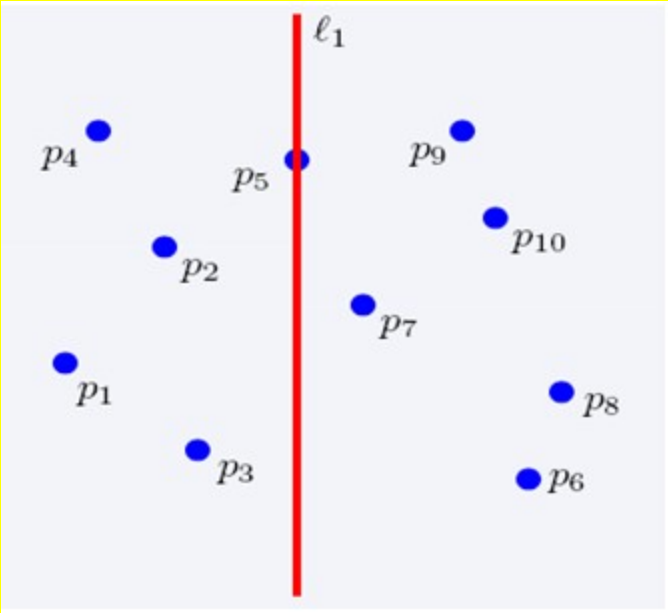

Fig-9 : Constructing kd-tree (step-2)

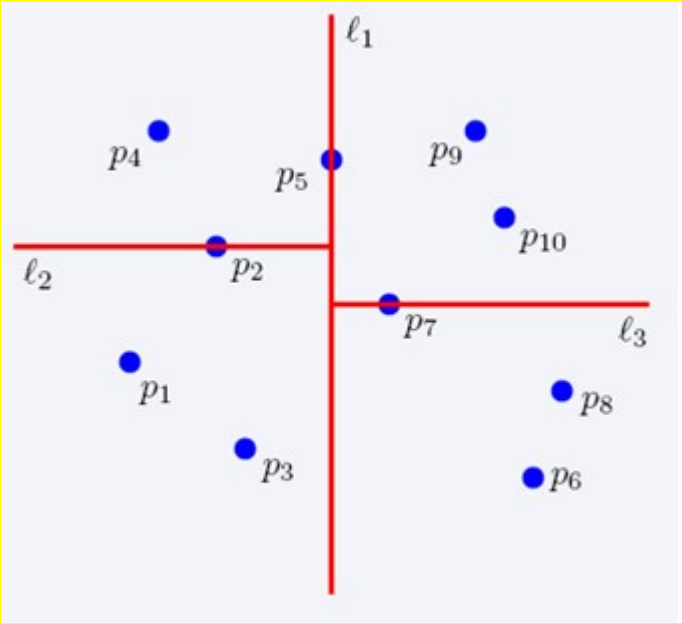

Fig-10 : Constructing kd-tree (step-3)

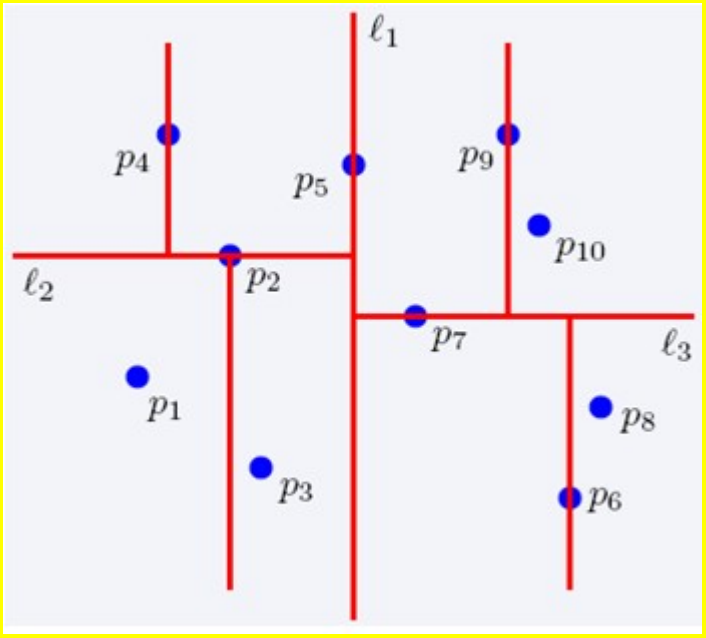

Fig-11 : Constructing kd-tree (step-4)

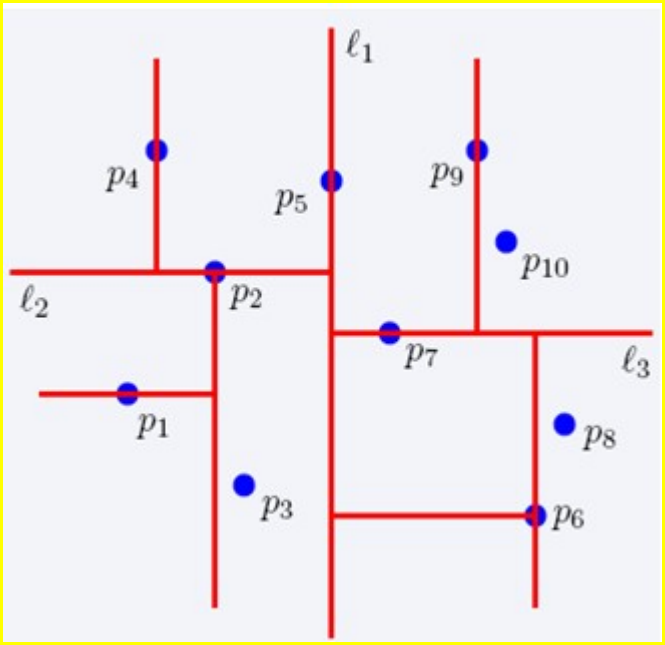

Fig-12 : Constructing kd-tree (step-5)

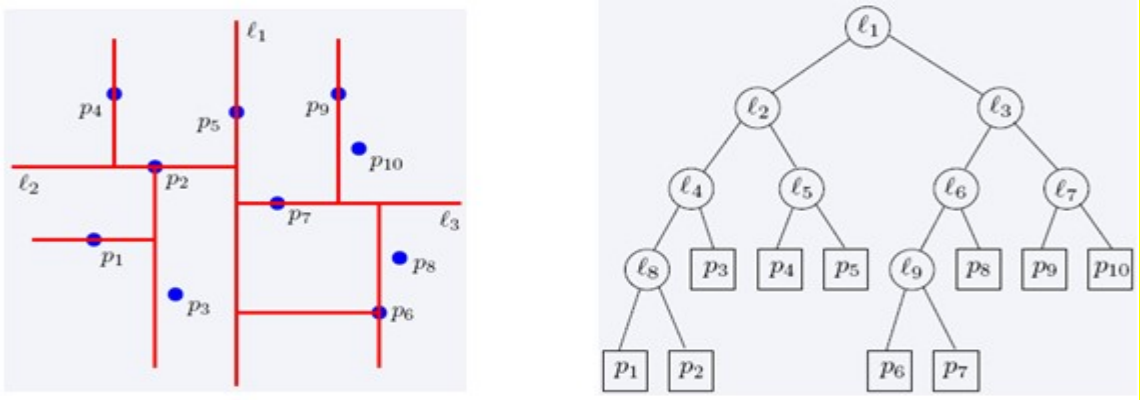

Fig-13 : Constructing kd-tree (step-6)

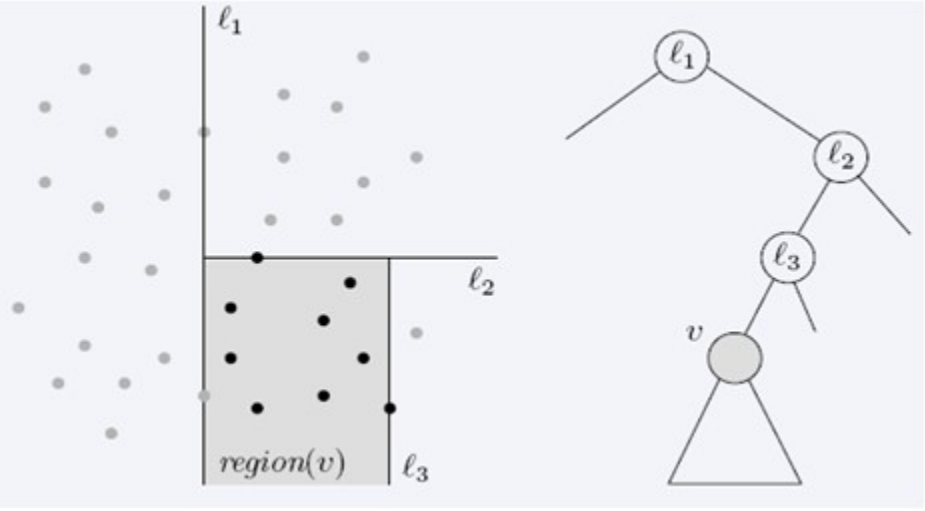

Fig-14 : Constructing kd-tree (step-7)

Q-5 : How kd-tree algorithm works in solving the problem of KNN classifier? What are the application of kd-tree?

The k-d tree (short for k-dimensional tree) algorithm is a data structure used to organize multidimensional data points in a hierarchical manner. It is commonly used to speed up the process of nearest neighbor search, which is essential in algorithms like the k-nearest neighbors (KNN) classifier.

Here's how the k-d tree algorithm works in solving the problem of the KNN classifier :

1. **Building the k-d Tree :** The first step is to build the k-d tree using the training data. The k-d tree recursively partitions the feature space into smaller regions by splitting it along the median of one of the dimensions. The dimension along which the split occurs alternates at each level of the tree.
2. **Nearest Neighbor Search :** When a new query point is given, the k-d tree is traversed to find its nearest neighbors. Starting from the root of the tree, the algorithm descends down the tree, choosing the branch that is closer to the query point at each level. The search terminates when a leaf node is reached.
3. **Backtracking :** After reaching a leaf node, the algorithm backtracks to explore other branches of the tree, ensuring that no closer neighbors exist in the unexplored regions. During backtracking, the algorithm may prune branches of the tree if it is guaranteed that they cannot contain closer neighbors than those already found.
4. **K Nearest Neighbors :** Once the nearest neighbors are found, the KNN classifier assigns the query point to the majority class among its k nearest neighbors.

The k-d tree algorithm speeds up the nearest neighbor search by efficiently partitioning the feature space and reducing the number of distance calculations required. Instead of

computing distances to all data points, the algorithm focuses on relevant regions of the feature space where nearest neighbors are likely to be found.

Applications of k-d trees include :

1. **K-Nearest Neighbors (KNN) Search :** K-d trees are widely used in KNN algorithms to efficiently find the nearest neighbors of a given query point in a high-dimensional space.
2. **Range Queries :** K-d trees can be used to efficiently perform range queries, where all points within a certain distance from a query point are retrieved.
3. **Image Processing :** In image processing applications such as image retrieval and image recognition, k-d trees can be used to accelerate the search for similar images based on their feature vectors.
4. **Spatial Databases :** K-d trees are used in spatial databases for indexing and querying multidimensional spatial data, such as geographic information systems (GIS) and location-based services.

Overall, k-d trees are versatile data structures that provide efficient solutions to nearest neighbor search problems in various domains, making them valuable tools in machine learning, data mining, and spatial data analysis.

Q-6 : For abnormal graphical type of data which type of problem Support Vector Machine faces? How to absorb the error? / What are the problems of Support Vector Machine in the given figure? How the problem can be solved?

Support Vector Machines (SVMs) are powerful supervised learning algorithms commonly used for classification and regression tasks. However, SVMs may face challenges when dealing with abnormal graphical types of data, such as data with imbalanced classes, noisy or overlapping clusters, or non-linear decision boundaries. Some specific problems that SVMs may encounter in such scenarios include :

1. **Imbalanced Classes :** When one class in the dataset is significantly more prevalent than the other(s), SVMs may struggle to learn an accurate decision boundary, as they tend to prioritize the larger class. This can lead to biased predictions and poor performance on the minority class.
2. **Noisy or Overlapping Data :** In cases where the classes are not well-separated or contain noisy data points, SVMs may produce suboptimal decision boundaries, resulting in misclassifications and reduced generalization performance.
3. **Non-Linear Decision Boundaries :** SVMs inherently assume linear separability between classes. When the underlying relationship between features and classes is

non-linear, SVMs may struggle to capture complex decision boundaries, leading to underfitting or overfitting.

To address these challenges and improve the performance of SVMs on abnormal graphical types of data, several strategies can be employed :

1. **Class Weighting :** Adjusting the class weights to penalize misclassifications of the minority class more heavily can help mitigate the effects of class imbalance and improve the SVM's ability to correctly classify minority instances.
2. **Kernel Tricks :** Utilizing non-linear kernels, such as polynomial kernels, radial basis function (RBF) kernels, or sigmoid kernels, can enable SVMs to capture complex, non-linear decision boundaries in the data.
3. **Feature Engineering :** Preprocessing the data by removing noise, reducing dimensionality, or transforming features using techniques such as principal component analysis (PCA) or feature scaling can help improve the separability of classes and enhance the performance of SVMs.
4. **Ensemble Methods :** Combining multiple SVM models trained on different subsets of the data or using different kernels can help improve generalization performance and robustness, especially in the presence of noisy or overlapping clusters.
5. **Cross-Validation :** Employing cross-validation techniques, such as k-fold cross-validation, can help assess the generalization performance of the SVM model and identify potential overfitting or underfitting issues.
6. **Regularization :** Tuning the regularization parameter (C) of the SVM model can help control the trade-off between maximizing the margin and minimizing classification errors, thus improving the model's ability to generalize to unseen data.

By employing these strategies, data scientists can enhance the performance and robustness of SVMs when dealing with abnormal graphical types of data, thereby improving the accuracy and reliability of classification outcomes.

Q-7 : Give an example of Naive Bayes Classifier. What is Bayes Theorem? What is the relation of Bayes Theorem with Naive Bayes Classifier? What is the difference between Bayes Theorem and Naive Bayes Classifier?

Let's solve a simple mathematical problem using Bayes' theorem :

Suppose we have two boxes, Box A and Box B, containing colored balls. Box A contains 3 red balls and 7 blue balls, while Box B contains 6 red balls and 4 blue balls. Now, we randomly select a box and then randomly select a ball from the chosen box. What is the probability that the selected ball is red?

We can solve this problem using Bayes' theorem. The Bayes Theorem states that, $P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$

Where,

- $P(A|B)$ is the probability of event A given event B has occurred.
- $P(B|A)$ is the probability of event B given event A has occurred.
- $P(A)$ is the prior probability of event A.
- $P(B)$ is the prior probability of event B.

In our problem :

- Let event A be selecting a red ball.
- Let event B be selecting a box (either box A or box B).

We want to find $P(\text{red ball}|\text{box})$, the probability of selecting a red ball given that a box has been chosen.

Let's denote $P(\text{red ball}) = P(A)$ and $P(\text{box}) = P(B)$

Now we need to calculate $P(\text{box}|\text{red ball})$, the probability of selecting a box given that a red ball has been chosen. This is given by : $P(\text{box}|\text{red ball}) = \frac{P(\text{red ball}|\text{box}) \times P(\text{box})}{P(\text{red ball})}$

We'll use the information provided to calculate these probabilities.

For Box A $P(\text{red ball}|\text{box A}) = \frac{3}{10}$ (Since Box A contains 3 red balls out of 10 total balls).

$P(\text{box A}) = \frac{1}{2}$ (Since there are 2 boxes and we select one randomely).

For Box B $P(\text{red ball}|\text{box B}) = \frac{6}{10}$ (Since Box B contains 6 red balls out of 10 total balls).

$P(\text{box B}) = \frac{1}{2}$ (Since there are 2 boxes and we select one randomely).

Now, we can calculate :

$$P(\text{box A}|\text{red ball}) = \frac{P(\text{red ball}|\text{box A}) \times P(\text{box A})}{P(\text{red ball})} = \frac{\frac{3}{10} \times \frac{1}{2}}{\frac{3+6}{10+10}} = \frac{1}{3}$$

$$P(\text{box B}|\text{red ball}) = \frac{P(\text{red ball}|\text{box B}) \times P(\text{box B})}{P(\text{red ball})} = \frac{\frac{6}{10} \times \frac{1}{2}}{\frac{3+6}{10+1}} = \frac{2}{3}$$

And, $P(\text{red ball}|\text{box}) = P(\text{red ball}|\text{box A}) \times P(\text{box A}) + P(\text{red ball}|\text{box B}) \times P(\text{box B})$

$= \frac{3}{10} \times \frac{1}{2} + \frac{6}{10} \times \frac{1}{2} = \frac{9}{20}$

So, the probability of selecting a red ball if any box is selected is $\frac{9}{20}$.

Bayes' theorem is a fundamental principle in probability theory that describes the probability of an event based on prior knowledge of conditions that might be related to the event. It is named after Thomas Bayes, an 18th-century British mathematician.

Mathematically, Bayes' theorem is expressed as : $P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$. Where :

- $P(A|B)$ is the conditional probability of event A occurring given that event B has occurred.
- $P(B|A)$ is the conditional probability of event B occurring given that event A has occurred.
- $P(A)$ and $P(B)$ are the probabilities of events A and B occurring, respectively.

Bayes' theorem provides a systematic way to update prior beliefs (expressed as probabilities) about an event based on new evidence.

The Naive Bayes classifier is a supervised learning algorithm based on Bayes' theorem and is particularly useful for classification tasks, especially in natural language processing tasks such as text classification and spam filtering.

The relation between Bayes' theorem and the Naive Bayes classifier lies in the application of Bayes' theorem to classify data. In the context of the Naive Bayes classifier, Bayes' theorem is used to calculate the probability that a given input belongs to a particular class based on the observed features of the input.

The key difference between Bayes' theorem and the Naive Bayes classifier lies in their scope and application :

- Bayes' theorem is a fundamental principle in probability theory that describes the relationship between conditional probabilities.
- The Naive Bayes classifier is a specific machine learning algorithm that uses Bayes' theorem to classify data, particularly in text classification tasks, by assuming that the features (i.e., the presence or absence of words in a document) are conditionally independent given the class label. This simplifying assumption makes the algorithm computationally tractable and often yields good results, especially in practice when the independence assumption holds approximately true.

In summary, Bayes' theorem provides the theoretical foundation for the Naive Bayes classifier, which is a practical application of the theorem in machine learning for classification tasks.

Lecture-7 : Decision Tree

Q-1 : How to determine the best split in Decision Tree? Which test condition is the best?

In decision tree algorithms, determining the best split involves selecting the feature and the threshold that maximizes the information gain or minimizes impurity. The best split

is the one that results in the most homogenous subsets of data after the split, leading to better predictive accuracy.

There are several methods for determining the best split in a decision tree :

1. **Information Gain (Entropy) :** Information gain measures the reduction in entropy (or uncertainty) achieved by splitting the data based on a particular feature. Entropy is a measure of impurity or disorder in a dataset. The feature and threshold that result in the highest information gain are chosen as the best split.
2. **Gini Impurity :** Gini impurity measures the probability of misclassifying a randomly chosen element if it were randomly labeled according to the distribution of labels in the subset. The feature and threshold that minimize the Gini impurity are selected as the best split.
3. **Reduction in Variance (for Regression Trees) :** For regression tasks, reduction in variance is used instead of information gain or Gini impurity. It measures the decrease in variance achieved by splitting the data based on a particular feature and threshold. The feature and threshold that result in the greatest reduction in variance are chosen as the best split.
4. **Chi-square Test (for Categorical Variables) :** In some cases, particularly when dealing with categorical variables, the chi-square test can be used to determine the significance of a split. The chi-square test evaluates whether the distribution of the target variable differs significantly between different categories of the input feature. If the test statistic exceeds a certain threshold (e.g., based on a significance level), the split is considered significant.

The choice of the best split criterion depends on the nature of the data and the specific problem at hand. Information gain and Gini impurity are widely used for classification tasks, while reduction in variance is commonly used for regression tasks. Chi-square test is useful when dealing with categorical variables.

In practice, decision tree algorithms often allow the user to specify the split criterion or automatically select the best criterion based on the problem type and dataset characteristics. Experimentation and validation with different split criteria can help determine which one works best for a particular problem.

Q-2 : Generate Decision Rule from Decision Tree.

To generate decision rules from a decision tree, we traverse the tree from the root node to the leaf nodes and extract the conditions at each node that lead to a classification decision. Each path from the root to a leaf node represents a decision rule. Here's a step-by-step process to generate decision rules from a decision tree :

1. **Start at the root node :** Begin at the root node of the decision tree.
2. **Traverse the Tree :** Traverse the tree from the root node to each leaf node, following the decision paths based on the conditions at each node.
3. **Extract Condition :** At each node, extract the conditions (split criteria) that determine the decision path. These conditions typically involve comparing feature values with threshold values.
4. **Combine Conditions :** Combine the conditions along the decision path to form a decision rule. Each decision rule consists of a set of conditions that, if satisfied, lead to a particular classification outcome.
5. **Repeat for Each Leaf Node :** Repeat steps 2-4 for each leaf node of the decision tree.
6. **Finalize Decision Rules (Optional) :** Once all decision paths have been traversed and decision rules extracted, finalize the decision rules by combining them into a concise, interpretable format.
7. **Prune Decision Rules (Optional) :** In some cases, decision trees may contain redundant or overly specific decision rules. Pruning techniques can be applied to simplify and generalize the decision rules while maintaining their predictive accuracy.

Here's an example to illustrate the process of generating decision rules from a decision tree :

Consider a decision tree for classifying whether a person will buy a product based on their age and income :

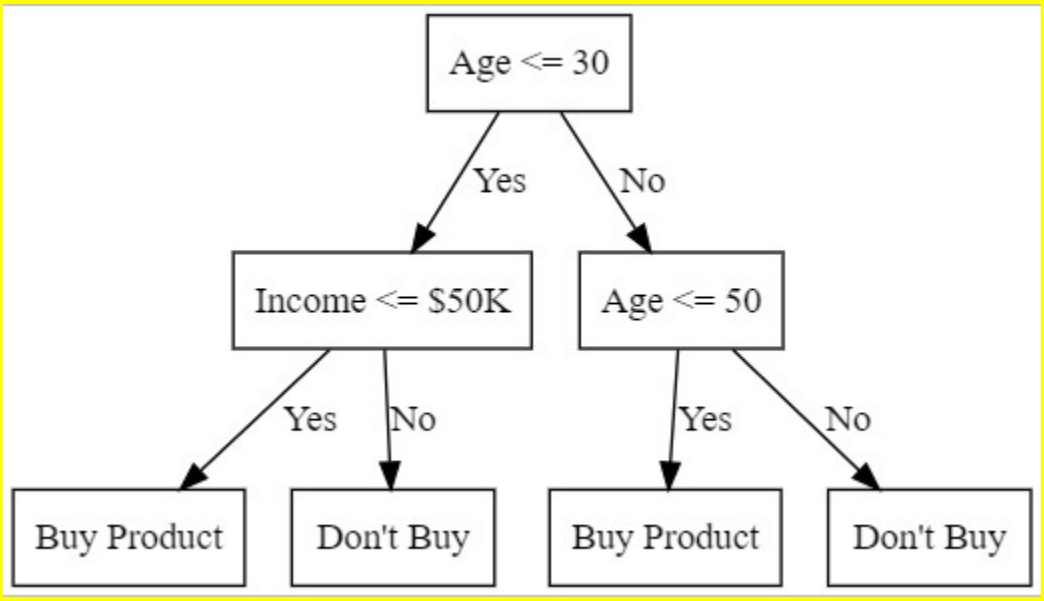

Fig-15 : Decision Tree-1

From this decision tree, we can generate the following decision rules :

1. If Age <= 30 and income <= $50K, then Buy Product.
2. If Age <= 30 and income > $50K, then Don't Buy.

3. If Age > 30 and Age<=50, then Buy Product.
4. If Age > 30 and Age > 50, then Don't Buy.

These decision rules provide clear and interpretable guidelines for predicting whether a person will buy a product based on their age and income.

Q-3 : Provide an example of Decision Tree and find out best route of Decision Tree.

let's consider a simple example of a decision tree for classifying whether to play tennis based on weather conditions. Here's the decision tree :

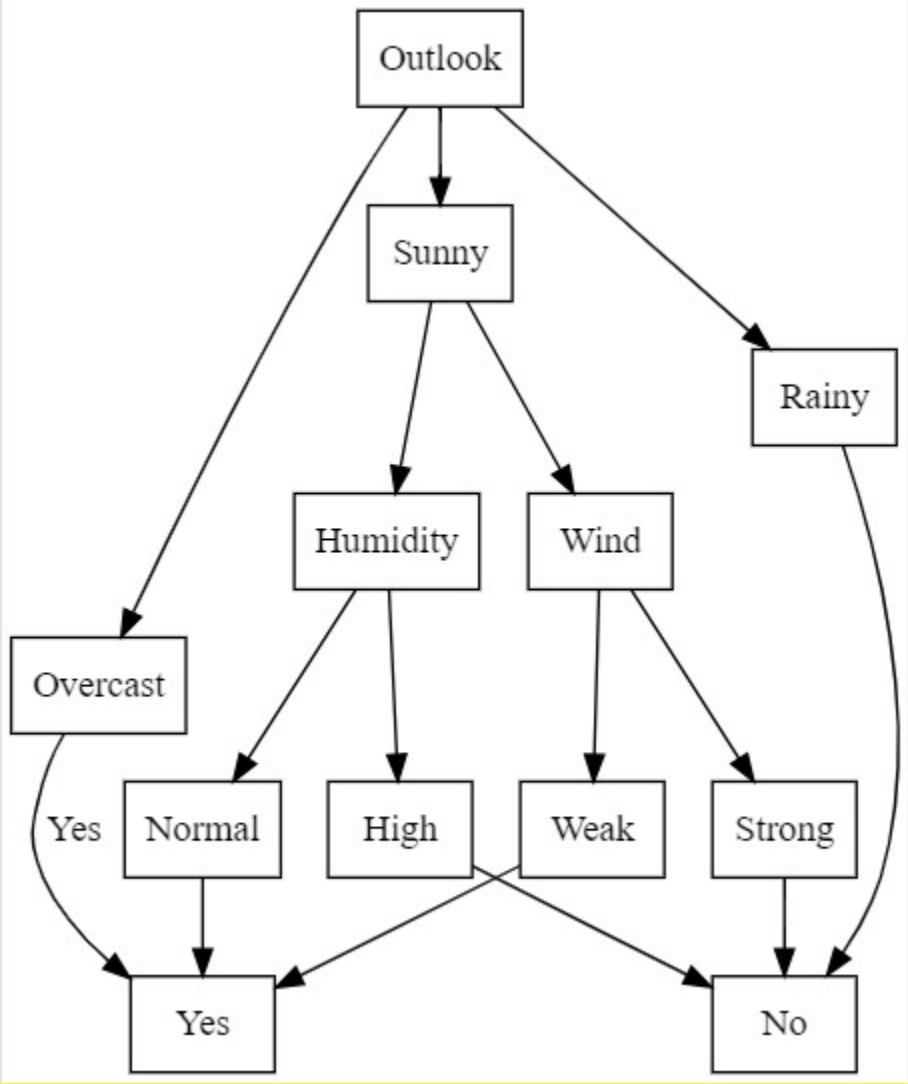

Fig-16 : Decision Tree-2

In this decision tree :

- The root node is "Outlook", which represents the outlook condition.
- There are three possible values for the Outlook: "Sunny", "Overcast", and "Rainy".
- Each internal node represents a decision based on a feature (e.g., humidity or wind).

- Each leaf node represents a classification decision (e.g., whether to play tennis or not).

Now, let's find out the best route (path) through the decision tree for a particular instance. For example, consider a day with the following conditions :

- Outlook : Sunny
- Humidity : Normal
- Wind : Weak

To find the best route through the decision tree for this instance, we start at the root node and follow the decision paths based on the conditions until we reach a leaf node. Here's the route through the decision tree :

1. Start at the root node "Outlook".
2. Since the outlook is "Sunny", follow the "Sunny" branch.
3. Next decision is based on "Humidity".
4. Since the humidity is "Normal", follow the "Normal" branch.
5. The next decision is based on "Wind".
6. Since the wind is "Weak", follow the "Weak" branch.

We have reached a leaf node with the decision "Yes" (to play tennis). So, the best route through the decision tree for the given instance is "Outlook (Sunny) -> Humidity (Normal) -> Wind (Weak) -> Play Tennis (Yes)".

This route represents the classification decision made by the decision tree for the given instance. It shows the sequence of decisions made at each node based on the instance's features, leading to the final classification outcome.

Q-4 : What is class attribute? Describe about the types of attribute.

A class attribute, also known as a target variable or dependent variable, is the variable in a dataset that we want to predict or classify. In supervised learning tasks, the class attribute is the variable that the model aims to predict based on the values of the input features. For example, in a dataset of housing prices, the class attribute might be the price of the house, while the input features could include factors such as the number of bedrooms, square footage, location, etc.

Now, let's describe the types of attributes commonly encountered in datasets :

1. **Nominal Attribute :** Nominal attributes are categorical variables with no inherent order or ranking among their values. They represent qualitative data where the categories are distinct and unordered. Examples of nominal attributes include :
   - Gender (Example : Male, Female)
   - Color (Example : Red, Blue, Green)

- Marital Status (Example : Single, Married, Divorced)

2. **Ordinal Attribute :** Ordinal attributes are categorical variables with a meaningful order or ranking among their values, but the intervals between the categories may not be uniform or quantifiable. They represent qualitative data where the categories have a natural order but the differences between them may not be consistent. Examples of ordinal attributes include :
   - Educational level (Example : High School, College, Bachelor's, Master's, PhD)
   - Rating scale (Example : Poor, Fair, Good, Excellent)
   - Socio-economic status (Example : Low, Middle, High)
3. **Continuous Attribute :** Continuous attributes are numerical variables that can take on any real value within a specified range. They represent quantitative data where the values are measurable and can be represented on a continuous scale. Examples of continuous attributes include :
   - Age (Example : 25, 30, 35)
   - Income (Example : $50,000, $75,000, $100,000)
   - Temperature (Example : 20°C, 25°C, 30°C)

Understanding the types of attributes in a dataset is crucial for selecting appropriate data preprocessing techniques and machine learning algorithms. Different types of attributes may require different handling during data preprocessing and may influence the choice of algorithms used for modeling. For example, nominal attributes may require one-hot encoding, ordinal attributes may require encoding with meaningful numerical values, and continuous attributes may require scaling to ensure consistent units and ranges.

Q-5 : Write the Decision tree learning algorithm (Optional).

```
Algorithm decisionTree(D, A, T)
1   if D contains only training examples of the same class c_j ∈ C then
2       make T a leaf node labeled with class c_j;
3   elseif A = ∅ then
4       make T a leaf node labeled with c_j, which is the most frequent class in D
5   else  // D contains examples belonging to a mixture of classes. We select a single
6         // attribute to partition D into subsets so that each subset is purer
7       p_0 = impurityEval-1(D);
8       for each attribute A_i ∈ {A_1, A_2, ..., A_k} do
9           p_i = impurityEval-2(A_i, D)
10      end
11      Select A_g ∈ {A_1, A_2, ..., A_k} that gives the biggest impurity reduction,
            computed using p_0 − p_i;
12      if p_0 − p_g < threshold then    // A_g does not significantly reduce impurity p_0
13        make T a leaf node labeled with c_j, the most frequent class in D.
14      else                             // A_g is able to reduce impurity p_0
15          Make T a decision node on A_g;
16          Let the possible values of A_g be v_1, v_2, ..., v_m. Partition D into m
                disjoint subsets D_1, D_2, ..., D_m based on the m values of A_g.
17          for each D_j in {D_1, D_2, ..., D_m} do
18              if D_j ≠ ∅ then
19                create a branch (edge) node T_j for v_j as a child node of T;
20                decisionTree(D_j, A-{A_g}, T_j)// A_g is removed
21              end
22          end
23      end
24  end
```

Fig-17 : Decision Tree Learning Algorithm