USN | | | | | | | | | | |

Internal Assessment Test 2 – July 2024

Sub:	Data Science and Visualization					Sub Code:	21CS644	Branch:	ISE
Date:	11/6/2024	Duration:	90 min's	Max Marks:	50	Sem/Sec:	VI / A, B		OBE

	Answer any FIVE FULL Questions	MARKS	CO3	L1
1 a)	Differentiate between Feature Generation and Feature Selection.	6	L3	L2
1 b)	Compare domain expertise versus machine learning algorithms.	4	L2	L2
2	Write the short notes on: i. Feature Selection Criteria. ii. Random Forest. iii. The three primary methods of regression iv. The Kaggle model.	10	L3	L2
3	Where and when should we use Pie Cart. Suppose you have data on the market share of different tech companies. Plot a pie chart using python showing the market share distribution among the companies, making it easy to see which company has the largest or smallest share.	10	L3	L3
4	Create a scatter plot using python to visualize the relationship between the age of used cars (in years) and their resale prices (in USD).	10	L1	L3

USN | | | | | | | | | | |

Internal Assessment Test 2 – July 2024

Sub:	Data Science and Visualization					Sub Code:	21CS644	Branch:	ISE
Date:	11/6/2024	Duration	90 min's	Max Marks	50	Sem/Sec	VI / A, B		OBE

	Answer any FIVE FULL Questions	Marks	CO	RBT
1 a)	Differentiate between Feature Generation and Feature Selection.	6	CO3	L2
1 b)	Compare domain expertise versus machine learning algorithms.	4	CO3	L2
2	Write the short notes on: 1. Feature Selection Criteria. 2. Random Forest. 3. The three primary methods of regression 4. The Kaggle model.	10	CO3	L2
3	Where and when should we use Pie Chart. Suppose you have data on the market share of different tech companies. Plot a pie chart using python showing the market share distribution among the companies, making it easy to see which company has the largest or smallest share.	10	CO4	L3
4	Create a scatter plot using python to visualize the relationship between the age of used cars (in years) and their resale prices (in USD).	10	CO4	L3

5		10 M	CO3	L3

Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Consider the above dataset based on which you will determine whether to play football or not using Decision Tree. Solve this using ID-3 Algorithm.

6 a)	List and discuss of comparison plots.	5 M	CO4	L2
6 b)	Differentiate between Line Chart and Scatter plot with an example of each	5 M	CO4	L2

Faculty Signature CCI Signature HOD Signature

5		10 M	CO3	L2

Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Consider the above dataset based on which you will determine whether to play football or not using Decision Tree example. Solve this using ID-3 Algorithm.

6 a)	List and discuss of comparison plots.	5 M	CO4	L3
6 b)	Differentiate between Line Chart and Scatter plot with an example of each	5 M	CO4	L1

Faculty Signature CCI Signature HOD Signature

SOLUTION

Ans- 1(a) Feature generation and feature selection are both essential steps in machine learning and data analysis, helping to improve model performance. However, they serve different purposes:

1. Feature Generation

Purpose: To create new features from existing data, often to capture patterns or interactions that raw features alone may not reveal.

Techniques:

Mathematical Transformations: Using functions (e.g., log, square root) to transform variables.

Feature Engineering: Combining features (e.g., product, ratio) to highlight relationships.

Dimensionality Reduction: Methods like Principal Component Analysis (PCA) create new features by compressing existing ones.

Domain Knowledge: Creating features based on expert knowledge of the dataset or industry. Outcome: New or modified features are added to the dataset, which may improve model prediction accuracy.

2. Feature Selection

Purpose: To choose the most relevant features from the dataset, reducing the number of inputs without losing critical information. This helps prevent overfitting, reduces computation, and gives useful features.

Ans-1(b) Domain expertise and machine learning algorithms represent two approaches for problem-solving and decision-making, and each has distinct strengths and limitations. Let's break down their roles, benefits, and potential drawbacks.

1. Domain Expertise

- **Definition**: Domain expertise is the knowledge and insights a person gains from experience and training in a specific field (e.g., finance, healthcare, engineering).
- **Strengths**:
    - **In-depth Knowledge**: Experts have a nuanced understanding of the field, including the context, complexities, and subtleties that may not be visible to outsiders.
    - **Critical Thinking**: Experts can make informed decisions based on intuition and experience, which can be particularly helpful when data is limited.
    - **Ethics and Judgement**: They can interpret outcomes within the ethical frameworks and societal norms specific to their field, making more responsible choices.
- **Limitations**:
    - **Bias and Subjectivity**: Personal biases and heuristics can sometimes affect their judgment.
    - **Scalability**: Human experts may struggle with processing large amounts of data quickly

Ans-2 1) **Feature Selection Criteria**

Feature selection is a process in machine learning that involves choosing the most relevant features (variables) from a dataset for model training to improve accuracy, reduce overfitting, and decrease computational time. Key criteria include:

- **Correlation**: Selecting features with a high correlation to the target variable and removing those highly correlated with each other.
- **Mutual Information**: Choosing features based on how much information they share with the target variable.
- **Statistical Tests**: Using statistical methods like chi-square for categorical data or ANOVA for continuous data to identify significant features.
- **Embedded Methods**: Leveraging methods like Lasso Regression that have built-in feature selection as part of model training.

- **Recursive Feature Elimination (RFE)**: Repeatedly fitting the model and removing the least important features to improve performance.

2) **Random Forest**

Random Forest is an ensemble machine learning algorithm primarily used for classification and regression. It builds multiple decision trees during training and combines their results to improve accuracy and control overfitting. Key aspects include:

- **Ensemble Learning**: Random Forest combines the predictions of multiple decision trees to reduce variance and improve generalization.
- **Bagging (Bootstrap Aggregating)**: Each tree is trained on a random sample of the data, making the model robust to noise.
- **Feature Randomness**: Each split considers only a subset of features, reducing correlation between trees and enhancing performance.
- **Feature Importance**: Random Forest can rank features by importance, aiding in feature selection.

3) **The Three Primary Methods of Regression**

Regression is a statistical technique for modeling the relationship between a dependent variable and one or more independent variables. The three primary methods are:

- **Linear Regression**: Models the linear relationship between variables, fitting a straight line to data points based on least-squares minimization.
- **Logistic Regression**: Used for binary classification, it models the probability of a categorical outcome, transforming predictions with a logistic function.
- **Polynomial Regression**: A form of linear regression where the model is extended with polynomial terms to capture non-linear relationships.

4) **The Kaggle Model**

Kaggle is a popular data science platform for model development and competition. Models developed for Kaggle typically adhere to a workflow that includes:

Data Exploration and Cleaning: Analyzing and preprocessing data to understand patterns and handle missing values.

Feature Engineering: Creating new features or transforming existing ones to better capture information relevant to the target variable.

Model Selection and Tuning: Testing various algorithms (e.g., XGBoost, Random Forest, Neural Networks) and optimizing hyperparameters to improve accuracy.

Evaluation: Using cross-validation and Kaggle's scoring metric to validate performance, ensuring the model generalizes well on unseen data.

Submission: Formatting the predictions according to Kaggle's requirements and submitting to see ranking on the leaderboard.

Ans-3 When and Where to Use Pie Charts: Pie charts are best used when you want to display the proportion of parts to a whole in a visually simple way. They work well for small datasets where each category is distinct and mutually exclusive. Pie charts are commonly used for showing market share, budget distribution, or survey results. However, they should be avoided if there are too many categories or if the differences between them are subtle, as this can make interpretation difficult.

```
import matplotlib.pyplot as plt

# Sample data: Market share of tech companies
companies = ['Company A', 'Company B', 'Company C', 'Company D']
market_share = [35, 25, 20, 20]

# Plotting the pie chart
plt.figure(figsize=(8, 8))
plt.pie(market_share, labels=companies, autopct='%1.1f%%', startangle=140)
plt.title("Market Share Distribution Among Tech Companies")
plt.show()
```

Ans -4 To create a scatter plot in Python that visualizes the relationship between the age of used cars and their resale prices, you would typically use the Matplotlib or Seaborn libraries. Below is a Python code example:

Assuming you have two lists, one for the age of cars and another for their resale prices:

```
import matplotlib.pyplot as plt

# Sample data: Age of cars and their resale prices
car_ages = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  # in years
resale_prices = [20000, 18000, 16000, 15000, 14000, 12000, 11000, 9000, 8000, 7000]  # in USD

# Plotting the scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(car_ages, resale_prices, color='b', marker='o')
plt.title("Relationship Between Age of Used Cars and Resale Prices")
plt.xlabel("Age of Cars (Years)")
plt.ylabel("Resale Prices (USD)")
plt.grid(True)
plt.show()
```

Ans- 5 Firtly, we are going to calculate the entropy for “Decision” attribute which is a target variable and also calculate the entropy for independent attributes like “Outlook”, “Temp.” , “Humidity” , “Wind” **.**

Decision	
Yes	9
No	5

Outlook								
Sunny	Yes	2	Overcast	Yes	4	Rain	Yes	3
	No	3		No	0		No	2

Temp.								
Hot	Yes	2	Mild	Yes	4	Cool	Yes	3
	No	2		No	2		No	1

Humidity					
High	Yes	3	Normal	Yes	6
	No	4		No	1

Wind					
Weak	Yes	6	Strong	Yes	3
	No	2		No	3

1)Entropy Calculation

Formula= $Entropy(S) = \sum - p(I) \cdot \log_2 p(I)$

We need to calculate entropy first.Decision column consists 14 instances includings two level Yes or no .There are 9 decision labeled yes and five decision labeled No.

Entropy(Decision)= -P(yes)*Log2(P(yes))-P(No)*log2(P(No))

Entropy(Decision) = 0.94029

Outlook= Sunny

It have Total 5 instances where 3 for Yes and 2 for No.

Entropy(Decision|Outlook=Sunny) = -P(No)*LOG2(P(No))-P(Yes)*LOG2(P(Yes))

Entropy(Decision|Outlook=Sunny) = 0.97095

Outlook=Overcast

It have total 4 instances where we have 4 Yes and 0 no.

Entropy(Decision|outlook=Overcast)= -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

Entropy(Decision|outlook=Overcast)= 0

Because this is a case of Pure split and there is no impurity present.

Outlook=Rain

It have 5 instances where we have 3 Yes and 2 No

Entropy(Decision|Outlook=Rain)= -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

Entropy(Decision|Outlook=Rain)= 0.97095

Information Gain

Information gain indicates how much information a particular feature gives us about the final outcomes.

Information Gain = **Gain(S, A) = Entropy(S) – ∑ [p(S|A) . Entropy(S|A)]**

Information Gain (Decision, Outlook)= Entropy(Decision) - ∑[P(Decision|Outlook) * Entropy(Decision|Outlook)]

Information Gain (Decision, Outlook)=

Entropy(Decision) - [P(Decision|Outlook=Sunny)*Entropy(Decision|Outlook=Sunny)]- [P(Decision|Outlook=Overcast)*Entropy(Decision|Outlook=Overcast)] - [P(Decision|Outlook=Rain)*Entropy(Decision|Outlook=Rain)]

0.27741

Temp

Temp=Hot

It have total 4 instances where we have 2 yes and 2 no.

Entropy(Decision|Temp=Hot) = -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

Entropy(Decision|Temp=Hot) = 1

Temp=Mild

It have total 6 instances where we have 4 yes and 2 no.

Entropy(Decision|Temp=Mild)= -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

Entropy(Decision|Temp=Mild)= 0.9183

Temp=Cool

It have total 4 instances where we have 3 Yes and 1 No.

Entropy(Decision|Temp=Cool)= -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

Entropy(Decision|Temp=Cool)= 0.81128

Information Gain

Information Gain (Decision, Temp)=

Entropy (Decision) -[P(Decision|Temp=Hot)*Entropy(Decision|Temp=Hot)] - [P(Decision|Temp=Mild)*Entropy(Decision|Temp=Mild)] - [P(Decision|Temp=Cool)*Entropy(Decision|Temp=Cool)]

Information Gain (Decision, Temp)= 0.02922

Humidity

Humidity=High

It have 7 instances where we have 3 yes and 4 No.

Entropy(Decision|Humidity=High) = -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

Entropy(Decision|Humidity=High) = 0.98523

Humidity=Normal

It have 7 instances where we have 6 yes and 1 No.

Entropy(Decision|Humidity=Normal)= -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

Entropy(Decision|Humidity=Normal)= 0.59167

Information Gain

Information Gain(Decision,Humidity)=
Entropy(Decision)-[P(Decision|Humidity=High)*Entropy(Decision|Humidity=High)]-[P(Decision|Humidity=Normal)*Entropy(Decision|Humidity=Normal)]

Information Gain(Decision,Humidity)= 0.15184

Wind

Wind=Weak

It have 8 instances where we have 6 yes and 2 No.

Entropy(Decision|Wind=Weak)= -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

Entropy(Decision|Wind=Weak)= 0.81128

Wind=Strong

It have 6 instances where we have 3 Yes and 3 No.

Entropy(Decision|Wind=Strong)= -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

Entropy(Decision|Wind=Strong)= 1

Information Gain

Information Gain(Decision,Wind)= Entropy(Decision)- [P(Decision|Wind=Weak)*Entropy(Decision|Wind=Weak)]-[P(Decision|Wind=Strong)*Entropy(Decision|Wind=Strong)]

Information Gain(Decision,Wind)= 0.04813

Information Gain

Information Gain (Decision, Outlook)=	0.27741
Information Gain (Decision, Temp)=	0.02922
Information Gain(Decision,Humidity)=	0.15184
Information Gain(Decision,Wind)=	0.04813

Here we can see that information gain is high for Outlook attributes so we can take this feature as the **Root Node.**

Repeat the same procedure on every branch until the decision node of each branch is finalized.

Now we need to test the Data set for the decision Node.

Outlook=Sunny

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

2)Entropy Calculation

Entropy(Outlook=Sunny)= -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

Entropy(Outlook=Sunny)= 0.97095

Outlook=Sunny and temp=Hot.

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No

Outlook=Sunny and Temp=Cool

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

Outlook=Sunny | Humidity=Normal

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Outlook=Strong | Wind=Strong

Day	Outlook	Temp.	Humidity	Wind	Decision
2	Sunny	Hot	High	Strong	No
11	Sunny	Mild	Normal	Strong	Yes

Outlook=Sunny | Wind=Weak

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes

Information Gain

Information gain(Outllok=Sunny|Temp)=	Entropy(Outlook=Sunny)-[P(Sunny|Temp=Hot)*Entropy(Sunny|Temp=Hot)]-[P(Sunny|Temp=Mild)*Entropy(Sunny|Temp=Mild)]-[P(Sunny|Temp=Cool)*Entropy(Sunny|Temp=Cool)]
Information gain(Outllok=Sunny|Temp)=	0.57095
Information gain(Outllok=Sunny|Humidity)=	Entropy(Outlook=Sunny)-[P(Sunny|Humidity=High)*Entropy(Sunny|Humidity=High)]-[P(Sunny|Humidity=Normal)*Entropy(Sunny|Humidity=Normal)]
Information gain(Outllok=Sunny|Humidity)=	0.97095
Information gain(Outllok=Sunny|Wind)=	Entropy(Outlook=Sunny)-[P(Sunny|Wind=Weak)*Entropy(Sunny|Wind=Weak)]-[P(Sunny|Wind=Strong]*Entropy(Sunny|Wind=Strong)]
Entropy(Sunny|Wind=Weak)=	-P(No)*log2(P(No))-P(Yes)*log2(P(Yes))
Entropy(Sunny|Wind=Weak)=	0.9183
Entropy(Sunny|Wind=Strong)=	-P(No)*log2(P(No))-P(Yes)*log2(P(Yes))
Entropy(Sunny|Wind=Strong)=	1
Information gain(Outllok=Sunny|Wind)=	0.01997

Here information gain for (Outlook=Sunny|Humidity) is high, so **Humidity** will be the next decision node.

Outlook=Overcast

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

Here decision will always be yes if outlook were overcast. So no need of calculating entropy and information gain.

Outlook=Rain

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Outlook=Rain | Temp=Mild

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Outlook=Rain |Temp=Cool

Day	Outlook	Temp.	Humidity	Wind	Decision
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No

Outlook=Rain | Humidity=High

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
14	Rain	Mild	High	Strong	No

Outlook=Rain | Humidity=Normal

Day	Outlook	Temp.	Humidity	Wind	Decision
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes

Outlook=Rain | Wind=Strong

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

Outlook=Rain | Wind=Weak

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

3)Entropy Calculation

Entropy(Outlook=Rain)= -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

Entropy(Outlook=Rain)= 0.97095

Information Gain

Information gain(Outlook=Rain|Temp)=

ntropy(Outlook=Rain)-[P(Rain|Temp=Hot)*Entropy(Rain|Temp=Hot)]-[P(Rain|Temp=Mild)*Entropy(Rain|Temp=Mild)]-[P(Rain|Temp=Cool)*Entropy(Rain|Temp=Cool

Entropy(Rain|Temp=Mild)= 0.9183

Entropy(Rain|Temp=Cool)= 1

Information gain(Outlook=Rain|Temp)= 0.01997

Information gain(Outllok=Rain|Humidity)=

ropy(Outlook=Rain)-[P(Rain|Humidity=High)*Entropy(Rain|Humidity=High)]-[P(Rain|Humidity=Normal)*Entropy(Rain|Humidity=Norm

Entropy(Rain|Humidity=High)= 1

ntropy(Rain|Humidity=Normal -P(No)*log2(P(No))-P(Yes)*log2(P(Yes))

ntropy(Rain|Humidity=Normal 0.9183

Information gain(Outllok=Rain|Humidity)= 0.01997

Information gain(Outllok=Rain|Wind)=

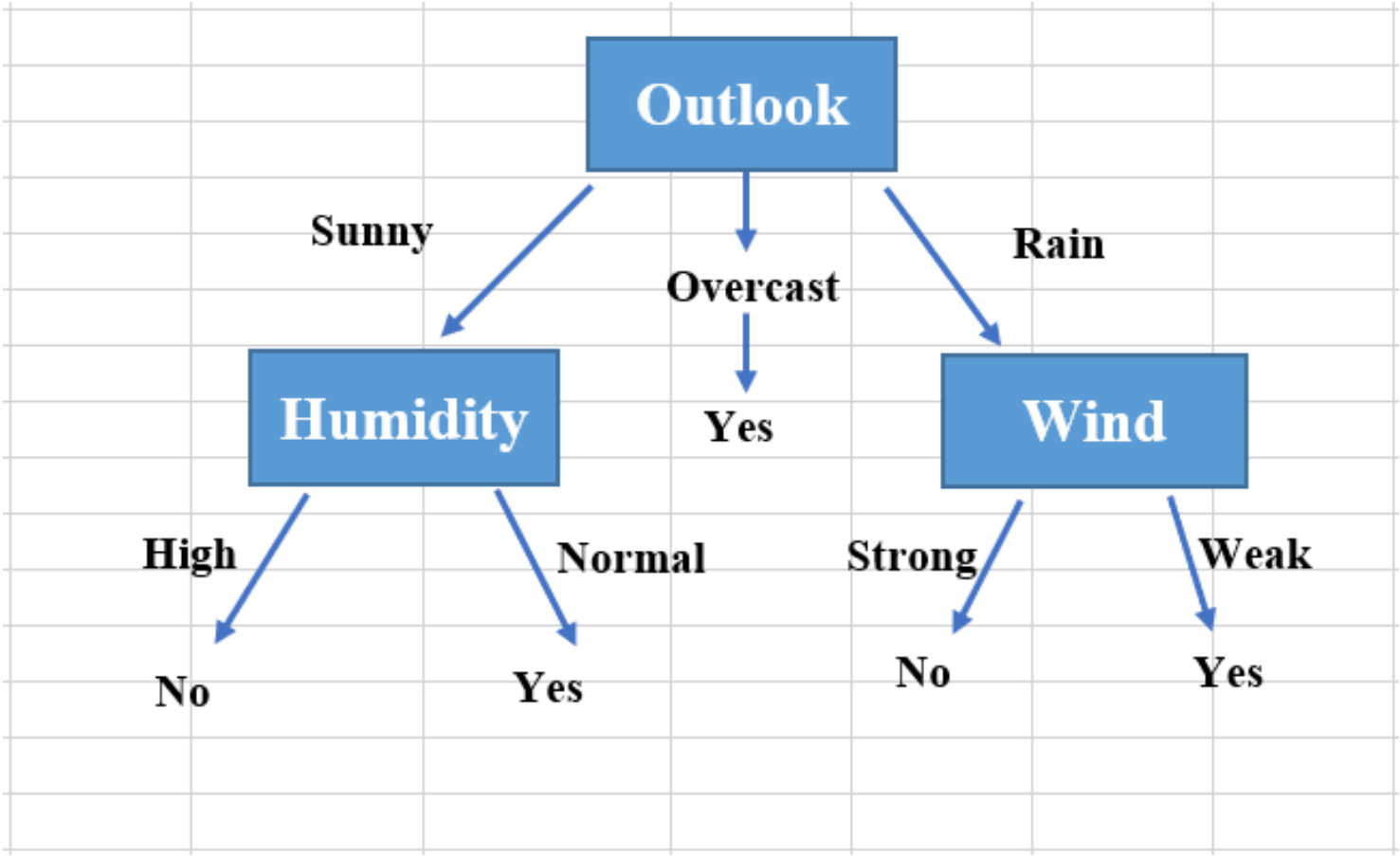
ntropy(Outlook=Rain)-[P(Rain|Wind=Weak)*Entropy(Rain|Wind=Weak)]-[P(Rain|Wind=Strong)*Entropy(Rain|Wind=Strong)

Entropy(Rain|Wind=Weak)= 0

Entropy(Rain|Wind=Strong)= 0

Information gain(Outllok=Rain|Wind)= 0.97095

Here we can see that information gain is high for (Outlook=Rain | Wind) , so it will be the decision node after Rain.



Decision Tree construction is over. Here we learned that how a decision tree is created in backend using this algorithm.

Ans-6(a)

- **Comparative Bar Graphs**:
  - **Description**: These graphs use bars to represent data values across different categories, allowing for easy comparison.
  - **Applications**: Useful in business for comparing sales figures, survey results, or demographic data.
- **Line Graphs**:
  - **Description**: Line graphs display information as a series of data points connected by straight line segments.
  - **Applications**: Commonly used to illustrate trends over time, such as stock prices, temperature changes, or population growth.
- **Scatterplots**:
  - **Description**: Scatterplots show individual data points plotted on a two-dimensional axis to depict the relationship between two variables.
  - **Applications**: Ideal for identifying correlations or patterns, such as the relationship between study time and exam scores.
- **Stacked Bar Graphs**:
  - **Description**: These are similar to comparative bar graphs but stack multiple data series on top of each other in a single bar.
  - **Applications**: Effective for showing the composition of data across categories, such as sales from different products over multiple years.
- **Comparative Advantage Graphs**:
  - **Description**: Typically used in economics, these graphs illustrate the concept of comparative advantage in trade, showing the opportunity costs of producing different goods.
  - **Applications**: Useful for demonstrating how countries or entities can benefit from specialization and trade based on their resource endowments.

Ans 6(b) Line Chart

Definition: A line chart displays data points connected by straight lines. It is typically used to represent trends over time or continuous data.

Usage:

- Ideal for showing how a value changes over time.
- Useful for displaying time series data where you want to see the movement of values.

Example: Imagine tracking the monthly sales of a company over a year. The x-axis represents the months (January to December), and the y-axis represents the sales figures. Each point on the line represents sales for that month, and the lines connect these points, showing the overall trend.

Scatter Plot

Definition: A scatter plot displays individual data points plotted on a Cartesian coordinate system. Each point represents two variables, with one variable on the x-axis and the other on the y-axis.

Usage:

- Ideal for showing relationships or correlations between two variables.
- Helps to identify patterns, trends, or clusters in the data.

Example: Consider a study investigating the relationship between the number of hours studied (x-axis) and exam scores (y-axis). Each point on the scatter plot represents a student, with their hours studied plotted against their exam score. This can help visualize if there's a correlation between study time and exam performance.

Key Differences

- Data Representation:
    - Line charts show trends over time with connected data points, while scatter plots display individual data points to show relationships between two variables.
- Interpretation:
    - Line charts are best for understanding changes and trends, whereas scatter plots are best for examining correlations and distributions.