CoGrammar

Welcome to this session:

Task Walkthrough - Tasks 17 - 21

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Skills Bootcamp Data Science

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: [Questions]

Skills Bootcamp Data Science

- For all **non-academic questions**, please submit a query: ***www.hyperiondev.com/support***
- **Report a safeguarding incident: *www.hyperiondev.com/safeguardreporting***
- We would love your feedback on lectures: ***Feedback on Lectures***
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

Learning Outcomes

- **Explain the differences** between Linear Regression, Logistic Regression, and Decision Trees.
- **Implement Linear Regression** to predict continuous outcomes.
- **Implement Logistic Regression** to classify categorical outcomes.
- **Apply Decision Trees** for classification and regression problems.
- **Compare and evaluate** the strengths, weaknesses, and use cases of each model.

Lecture Overview

→ Presentation of the Task
→ Machine Learning
→ Linear Regression
→ Logistic Regression
→ Decision Trees
→ Task Walkthrough

CoGrammar

Task Walkthrough

Imagine you are a data scientist in a healthcare organization. Your team is developing a model to predict diabetes risk and glucose levels based on patient information. Your tasks include:

- ❖ Linear Regression model: to predict Glucose Levels using BMI, Age, and BP.
- ❖ Logistic Regression model: to classify Diabetes Risk (Yes/No).
- ❖ Decision Tree model: to classify patients into risk categories based on their features.
- ❖ Compare model performance and decide which one is best for this problem.

What type of variable does Linear Regression predict?

A. Categorical
B. Numerical
C. Boolean
D. Ordinal

CoGrammar

When is Logistic Regression used instead of Linear Regression?

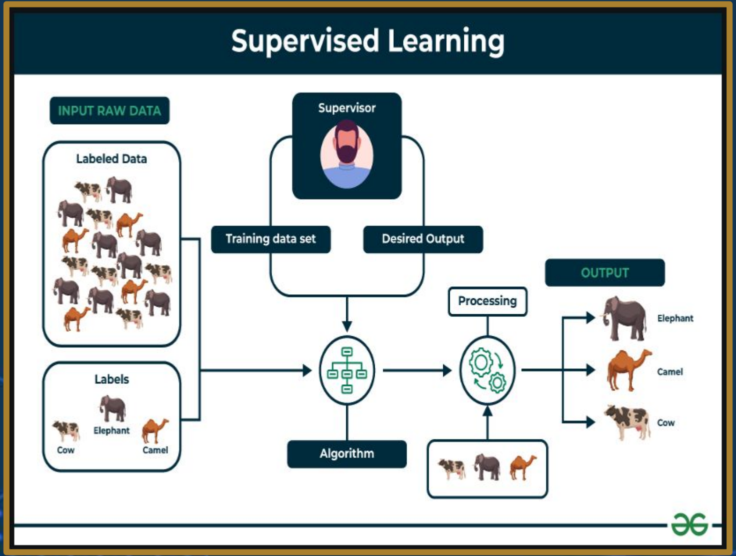
A. When predicting numerical values
B. When classifying categorical outcomes
C. When performing clustering analysis
D. When visualizing data

Introduction to Machine Learning

- Machine learning is a way of teaching computers to learn and improve from experience without being explicitly programmed.
- It allows computers to automatically learn and adapt based on data.

Types of machine learning

❖ **Supervised learning:** The computer learns from labelled data, where both input and output data are provided.

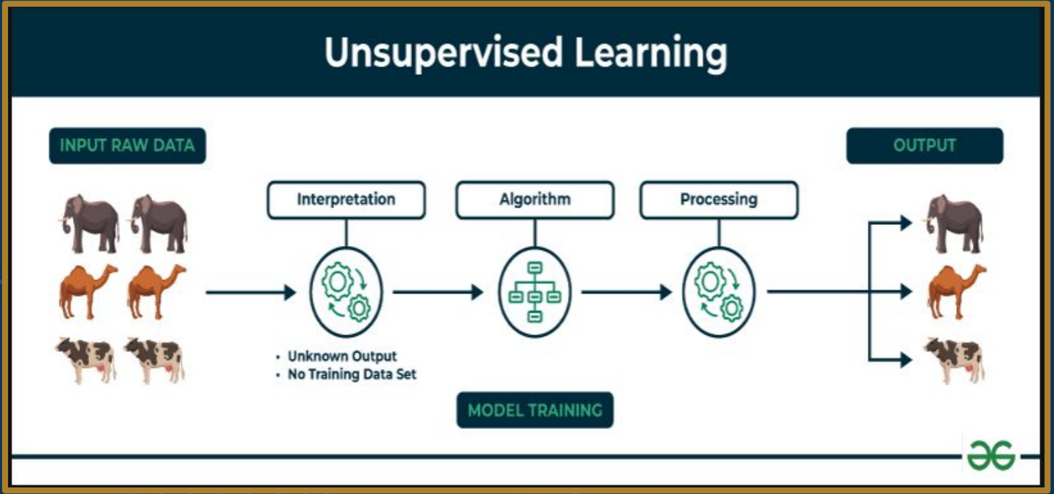


Source: [geeksforgeeks]

CoGrammar

Types of machine learning

- **Unsupervised learning:** The computer learns from unlabeled data, discovering hidden patterns or structures on its own.

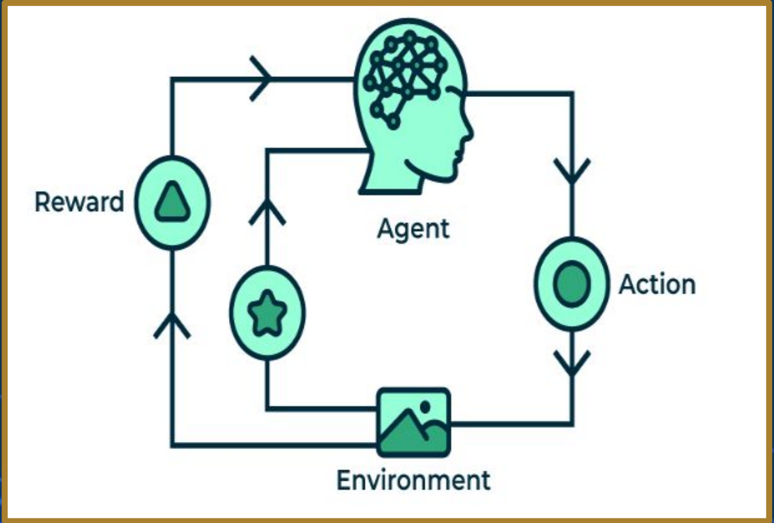


Source: geeksforgeeks

CoGrammar

Types of machine learning

- **Reinforcement learning:** The computer learns through interaction with an environment, receiving rewards or penalties for its actions.



Source: [geeksforgeeks]

CoGrammar

Types of Supervised Learning

- **Regression:** Predicting continuous numerical values, such as house prices or stock prices.
- **Classification:** Predicting discrete categories or classes, such as whether an email is spam or not.

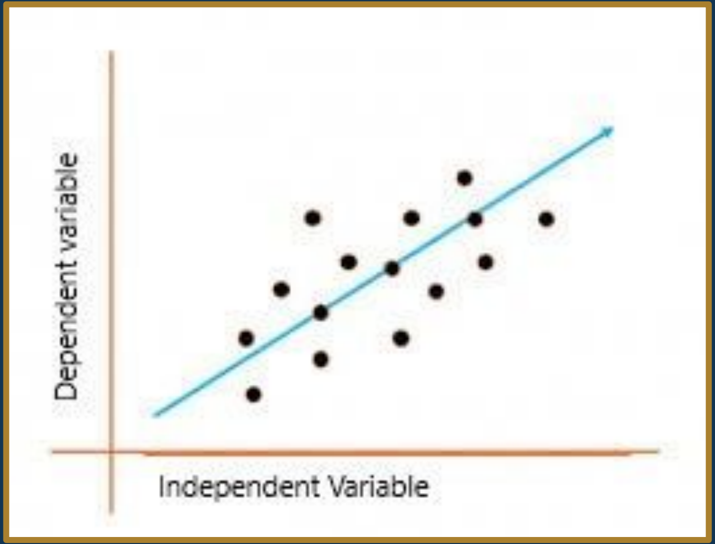
CoGrammar

Supervised Learning Algorithms

- **Linear regression:** Fitting a straight line to data points to make predictions.
- **Logistic regression:** Predicting binary outcomes, such as yes/no or true/false.
- **Decision trees:** Making decisions based on a series of questions or conditions.
- **Support vector machines (SVM):** Finding the best boundary to separate different classes.
- **Neural networks:** Mimicking the structure and function of the human brain to learn complex patterns.

Simple Linear Regression

- Simple linear regression is a method to study the relationship between two variables: an independent variable (x) and a dependent variable (y).
- It helps us understand how changes in the independent variable affect the dependent variable.



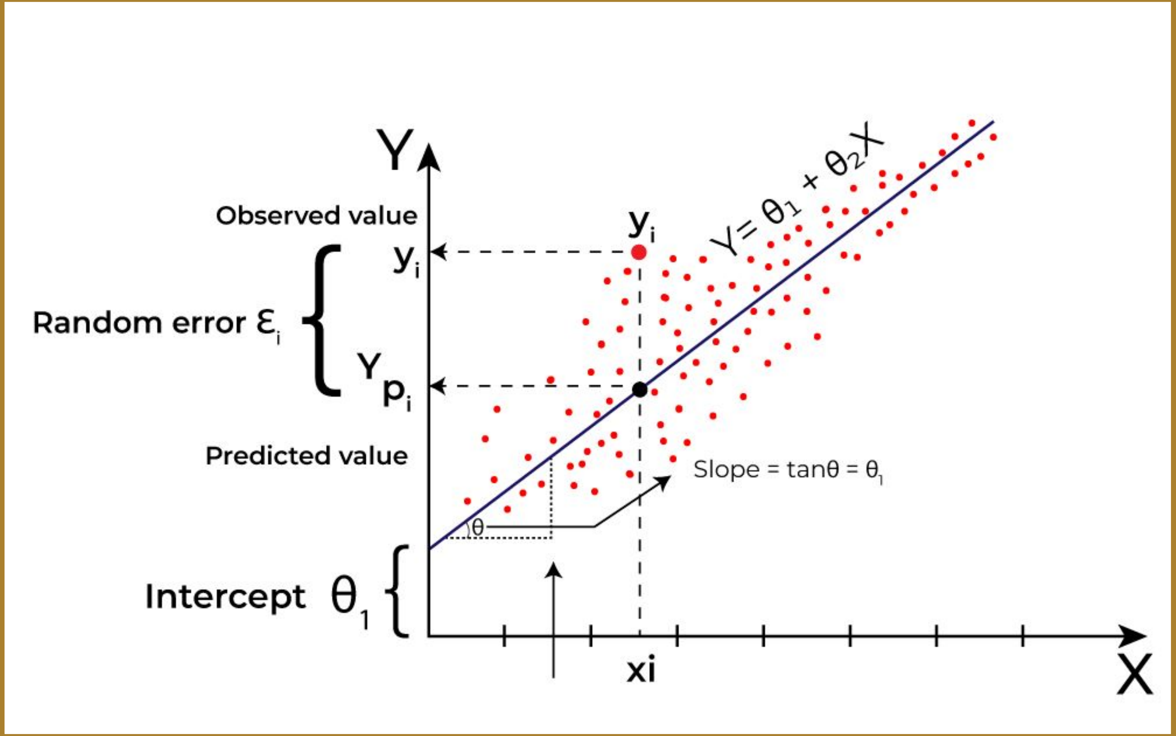
Source: Analytics Vidhya

CoGrammar

Math behind Simple Linear Regression

- The equation is written as: $y = \beta_0 + \beta_1 x + \varepsilon$
  - β_0 is the intercept, representing the value of y when x is zero.
  - β_1 is the slope, indicating how much y changes for a one-unit increase in x.
  - ε is the error term, accounting for the variability in y that cannot be explained by x.

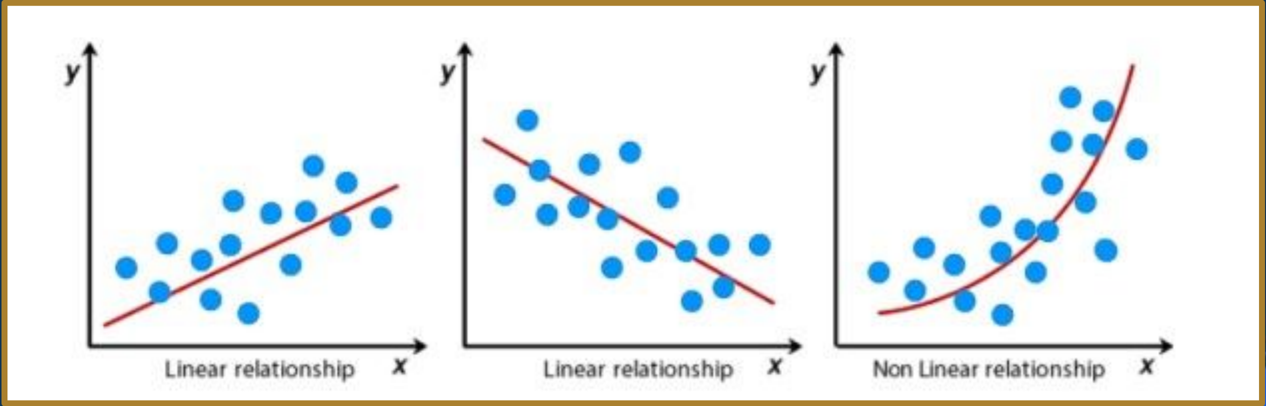
CoGrammar



Source: geeksforgeeks

Assumptions and Limitations of Simple Linear Regression

- **Linearity:** The relationship between x and y should be linear.

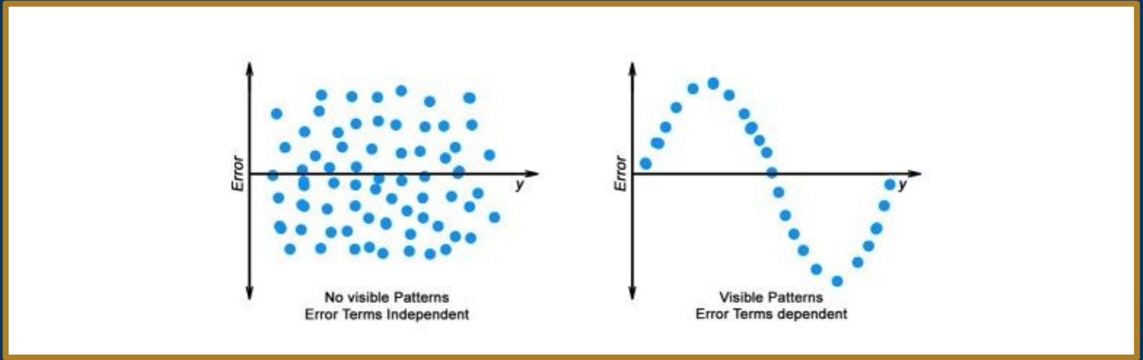


Source: [Analytics Vidhya]

CoGrammar

Assumptions and Limitations of Simple Linear Regression

❖ **Independence:** The observations should be independent of each other.

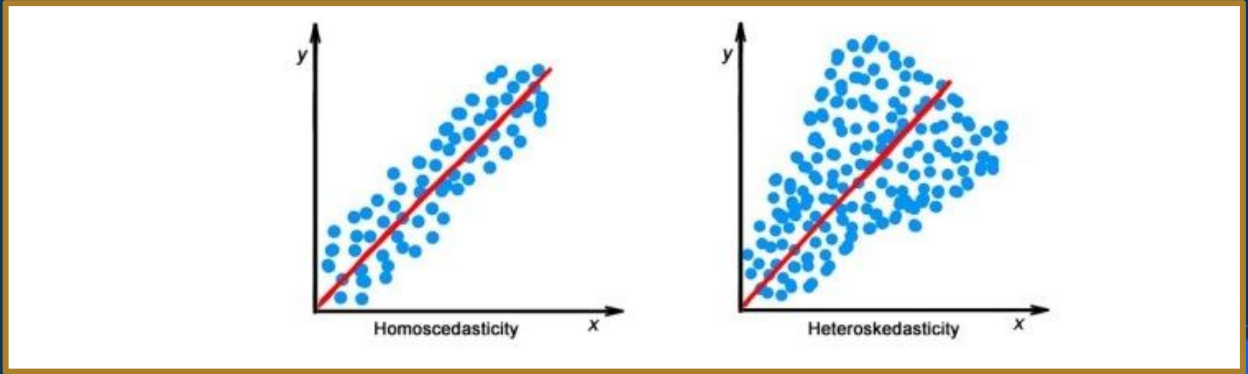


Source: [Analytics Vidhya]

CoGrammar

Assumptions and Limitations of Simple Linear Regression

❖ **Homoscedasticity:** The variability of y should be constant across all values of x.

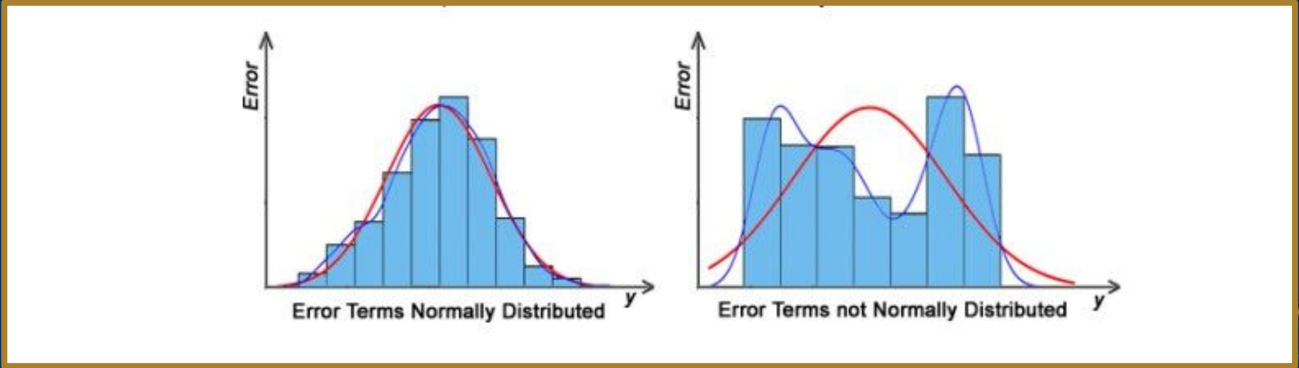


Source: [Analytics Vidhya]

CoGrammar

Assumptions and Limitations of Simple Linear Regression

❖ **Normality:** The errors should be normally distributed.



Source: [Analytics Vidhya]

CoGrammar

Scikit-learn

- Scikit-learn is a popular Python library for machine learning.
- It provides simple and efficient tools for data analysis and modelling.

```python
from sklearn.datasets import load_diabetes
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

CoGrammar

Evaluation Metrics

- Mean Squared Error (MSE):
  - MSE measures the average squared difference between the predicted and actual values.
  - A lower MSE indicates better model performance.
- R-squared (R^2) score:
  - R^2 represents the proportion of variance in the target variable that can be explained by the model.
  - An R^2 value closer to 1 indicates a better fit of the model to the data.

CoGrammar

Evaluation Metrics

- **Accuracy** is another commonly used metric for evaluating the performance of a machine learning model, particularly in classification problems.
  - **Accuracy = (Number of correct predictions) / (Total number of predictions) * 100%**
- While accuracy is more suitable for classification tasks, metrics like Mean Squared Error (MSE) and R-squared (R^2) are used for regression problems.

CoGrammar

Logistic Regression

- **Linear regression** models make **predictions** for the datasets for which dependent variables have **continuous numerical values.**

- **Logistic Regression**
  - **supervised learning** algorithm
  - **classification** algorithm
  - dependent variables are **distinct, non-continuous, categorical**

- **Classification** - predicting **probability** of **categorical variables** for a given observation and assigning the observation to the category with the highest probability.

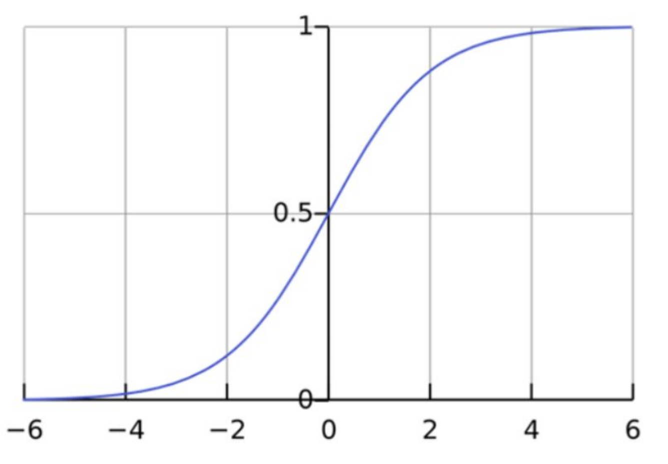
CoGrammar

Logistic function

- Logistic regression: statistical model that uses the **logistic (logit) function**, as the equation between x and y (also called **sigmoid function** or **S-shaped curve**).

- Returns only values between 0 and 1 for the dependent variable, irrespective of the values of the independent variable.

- Also model equations between **multiple independent variables** and **one dependent variable.**



Sigmoid function

$$p = \frac{1}{(1 + e^{-y})}$$

CoGrammar

Assumptions of Logistic Regression

- The **independent variables** should **not be correlated** with each other i.e. the model should have little or **no multicollinearity**.
- The **dependent variable** must be **categorical** in nature.
- The relationship between the **independent variables** and the **log odds** of the dependent variable should be **linear**.
- There should be **no outliers i**n the dataset.
- The data sample size should be **sufficiently large.**

CoGrammar

Data splitting

```python
# Splitting the dataset into features and target
X = data.drop('smoker', axis=1)
y = data['smoker']

# Splitting the data into training and testing sets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Normalize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

CoGrammar

Model fitting and prediction

```python
# Fit the logistic regression model
from sklearn.linear_model import LogisticRegression

log_reg = LogisticRegression()
log_reg.fit(X_train_scaled, y_train)
```

```python
# Predict the model
y_pred = log_reg.predict(X_test_scaled)
```

CoGrammar

Accuracy

Accuracy of classifier: Total number of correct predictions by the classifier divided by the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

For virus example, **Accuracy = 96%**

According to the **Accuracy** value, the model "can predict sick people 96% of the time". However, it is **predicting the people who will not get sick with 96% accuracy while the sick are spreading the virus.**

Better to measure how many **positive cases we can predict correctly** to arrest spread of the contagious virus or **out of the correct predictions**, how many **are positive cases** to check the reliability of the model.

CoGrammar

Precision and Recall

- **Precision:** tells us how many of the correctly predicted cases actually turned out to be positive, determine whether the model is reliable or not.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall**: how many of the actual positive cases we were able to predict correctly with our model.

$$Recall = \frac{TP}{TP + FN}$$

For virus example, Precision = 50%, Recall = 75%

For virus example, 50% percent of the correctly predicted cases turned out to be positive cases. Whereas 75% of the positives were successfully predicted by the model.

Precision and Recall

- **Precision**: useful in cases where **False Positive** is a greater concern.
- *Music* or *video recommendation systems, e-commerce websites.*
- *Wrong results* could lead to customer churn and be *harmful* to the business.

- **Recall:** useful in cases where **False Negative** trumps.
- *Medical cases* where it does not matter whether a false alarm flag is raised, but the *actual positive cases should not go undetected*.

For **contagious virus example**, the **Confusion Matrix** is more insightful measure in such critical scenarios.

Recall, assessing the ability to capture all actual positives, emerges as a **better metric**. **Accuracy** proves **inadequate** as a metric for the model's evaluation.

Avoid mistakenly releasing an infected person into the healthy population, potentially spreading the virus.

CoGrammar

F1-score

- Cases where there is no clear distinction between whether Precision is more important or Recall.

- **F1-score:** harmonic mean of **Precision** and **Recall**, gives a combined idea about these two metrics, appropriate metric for imbalanced dataset.

- **Maximum** when **Precision** is **equal** to **Recall**.

- Use in combination with other evaluation metrics.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

CoGrammar

Metrics using scikit-learn

```
#Evaluate the model
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

print("Accuracy Score:",accuracy_score(y_pred, y_test))
print("Confusion Matrix: \n",confusion_matrix(y_pred, y_test))
print("Classification Report: \n " ,classification_report(y_pred, y_test))
```

Classification report: Precision, Recall, and F1-score for each target class.

Macro average = average of Precision / Recall /F1-score.

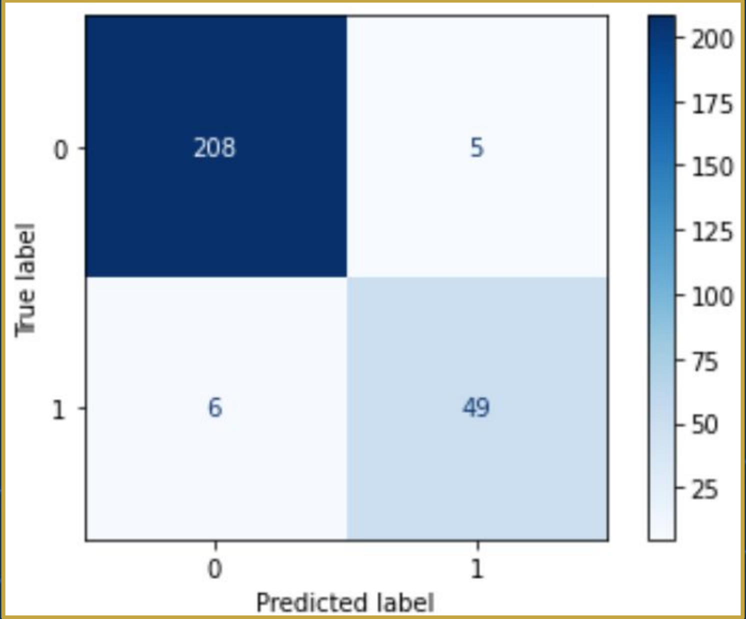
Weighted average of Precision / Recall / F1-score.

```
Accuracy Score: 0.9589552238805971
Confusion Matrix:
 [[208   5]
 [  6  49]]
Classification Report:
               precision    recall  f1-score   support

           0       0.97      0.98      0.97       213
           1       0.91      0.89      0.90        55

    accuracy                           0.96       268
   macro avg       0.94      0.93      0.94       268
weighted avg       0.96      0.96      0.96       268
```

CoGrammar

Metrics using scikit-learn

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm = confusion_matrix(y_pred, y_test,labels=log_reg.classes_)
# sns.heatmap can also be used to get the confusion matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=log_reg.classes_)
disp.plot(cmap='Blues')
```



Decision Trees

- Represent data by **partitioning** it into different **smaller subsets** based on questions asked of predictive variable in the data.

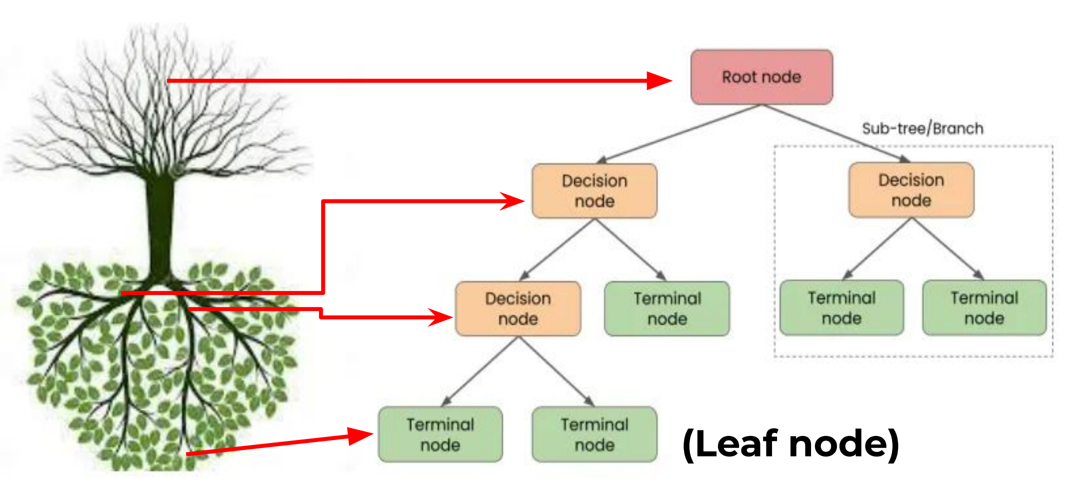
- **Hierarchical**: model is defined by a **sequential questions** that lead to a class label or a value when applied to any observation; model acts like a protocol in a series of “if this occurs then this occurs” conditions that produce a specific result from input data.

- **Non-parametric**: model is constructed based on the observed data; there are no underlying assumptions about the distribution of the errors or the data.

Components of Decision Trees

- **Root Node:** initial node at the beginning of a decision tree, where the entire population or dataset starts dividing based on various features or conditions.

- **Decision Nodes:** nodes resulting from the splitting of root nodes; represent intermediate decisions or conditions within the tree.



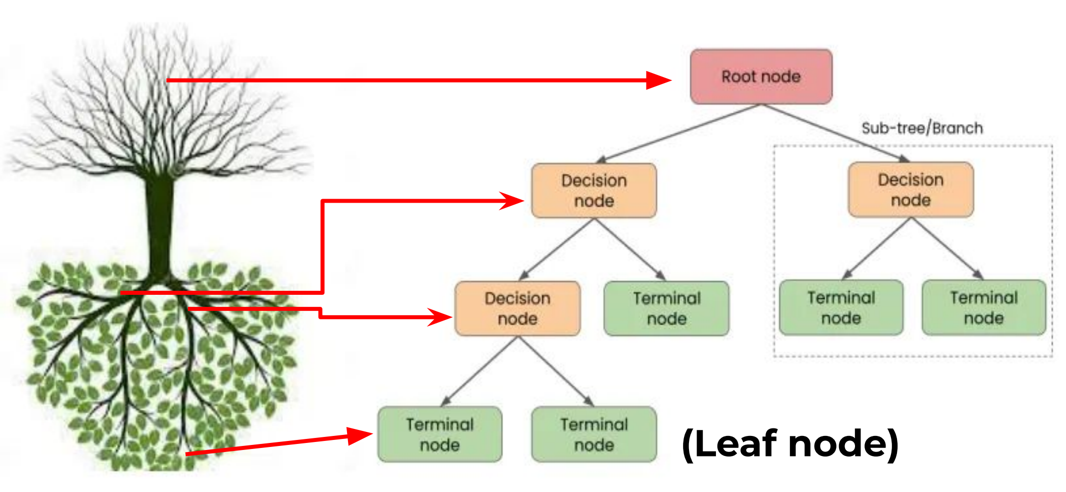
- **Leaf (Terminal) Nodes:** nodes where further splitting is not possible, often indicating the final classification or outcome.

HyperionDev

Components of Decision Trees

- **Branch / Sub-Tree:** subsection of entire decision tree; represents specific path of decisions and outcomes within the tree.

- **Parent and Child Node**: **Parent node** is divided into sub-nodes or **child nodes**. Parent represents a decision or condition. Child nodes represent outcomes or further decisions.



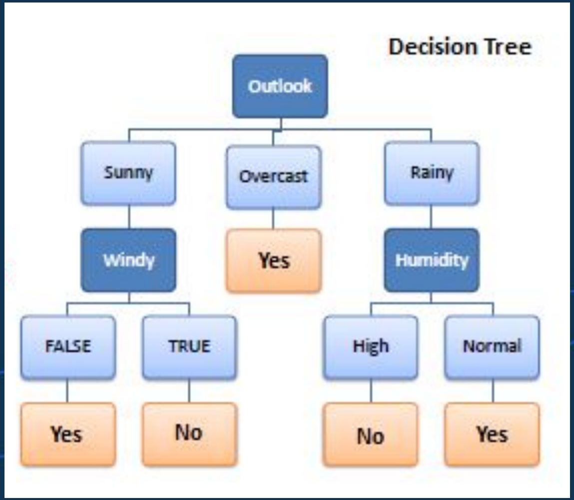
- **Pruning**: The process of removing or cutting down specific nodes in a decision tree to prevent overfitting and simplify the model.

HyperionDev

Classification Trees

Decision tree models where the **target variable** uses a **discrete set of values**, **classification** problems, determine whether an event happened or didn't happen, involving a “yes” or “no” outcome. Each **node,** or **leaf**, represent **class labels** while **branches** represent conjunctions of **features** leading to class labels.

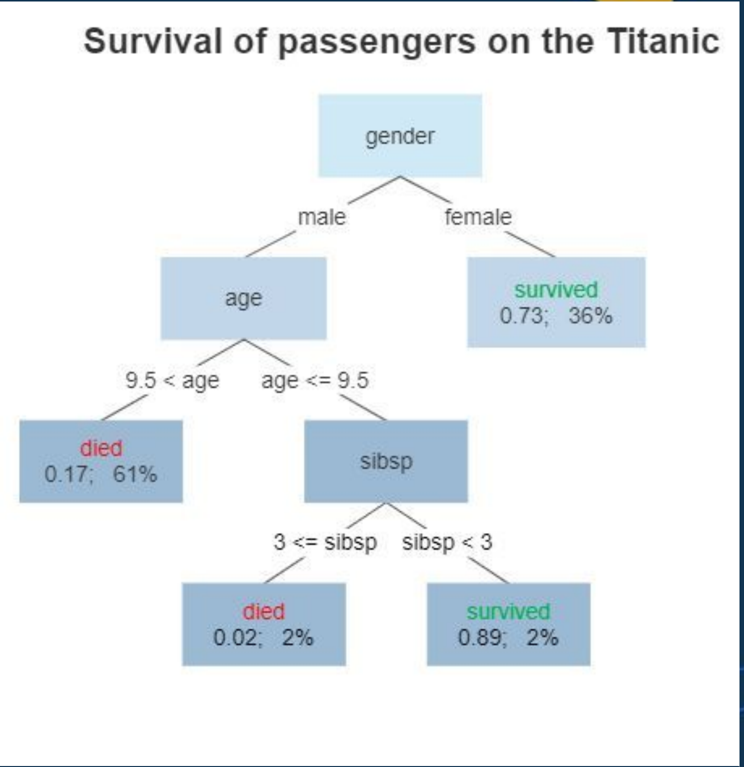
- The **root node (Outlook)** has two or more **decision nodes (Sunny, Overcast and Rainy)** with other **predictors** **(Windy, Humidity)**.
- The **leaf node** (**Play golf**) is the **target**, and represents a classification of decision.



Regression Trees

Decision trees which **predict continuous values** as **targets** based on previous data or information sources. **Predicts** what is likely to happen, given previous behavior/trends.

- Survival of passengers on the Titanic. Figures under the leaves show the **probability of survival** and the **percentage of observations** in the leaf.
- "sibsp" is the number of spouses or siblings aboard.



HyperionDev

If only interested in **whether 'survived' or 'died',** we can use a **classification tree** instead.

CART algorithm

Classification and Regression Trees (CART) algorithm

- **Tree structure:** CART builds a tree-like structure with nodes and branches.
- **Nodes**: represent different decision points.
- **Branches:** represent possible outcomes.
- **Leaf nodes:** contain a predicted class label or value for the target variable.
- **Splitting criteria:** CART evaluates all possible splits and selects the one that best reduces the impurity of the resulting subsets.
  - **Gini impurity** (for **classification**, lower means purer subset) and **residual reduction** (for **regression**, lower means better model's fit to the data).
- **Pruning:** done to prevent overfitting of the data, removes the nodes that contribute little to the model accuracy.

Task Walkthrough

Imagine you are a data scientist in a healthcare organization. Your team is developing a model to predict diabetes risk and glucose levels based on patient information. Your tasks include:

- Linear Regression model: to predict Glucose Levels using BMI, Age, and BP.
- Logistic Regression model: to classify Diabetes Risk (Yes/No).
- Decision Tree model: to classify patients into risk categories based on their features.
- Compare model performance and decide which one is best for this problem.

What does the Sigmoid function in Logistic Regression do?

A. Predicts continuous values
B. Normalizes data between 0 and 1
C. Converts a linear output into a probability
D. Increases model accuracy

What makes Decision Trees different from Regression models?

A. They require fewer features
B. They work only with categorical data
C. They split data into branches based on conditions
D. They cannot be visualized

Summary

★ **Linear Regression** predicts continuous values.

★ **Logistic Regression** classifies binary outcomes.

★ **Decision Trees** handle both classification and regression with clear decision rules.

CoGrammar

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education