Artificial Intelligence-Curriculum

1.INTRODUCTION TO AI

1.1 Overview of AI

- The simulation of human intelligence processes by machines, especially computer systems

1.2 Applications of Al

- Healthcare
- Finance
- Autonomous Vehicles
- Robotics
- Education
- Gaming

1.3 Al Foundation and History

- Mathematics
- Linguistics
- Control Theory & Cybernetics
- Computer Engineering

1.4 Types of Al

- Capabilities - Weak AI, General AI, Super AI
- Functionalities - Relative Machines, Limited Memory, Theory of Mind

2.PYTHON

2.1 Introduction to Python

2.2 Python Functions, Packages, and Routines.

Functions

- Fuctions are blocks of reusable code that perform a specific task. They are defined using the def keyword, allow parameters, and can return results, making code more modular and organised.

Python Packages

- Packages are collections of modules that group related functions, classes, and routines together.

Routines

- Refers to a series of programmed instructions or functions that can be reused to perform common tasks. They help automate processes, improve efficiency, and minimise code duplication.

2.3 Data Types, Operators, Variables

Data Types

- Python supports various data types, including integers (int), floating-point numbers (float), strings (str), and complex types like lists, tuples, dictionaries, and sets for managing diverse kinds of data.

Operators

- Python provides operators for performing operations on variables and values, including arithmetic (+, -, *, /), comparison (==,!=, <, >), logical (and, or, not), and assignment (=, +=, -=) operators.

Variables

- Variables are symbolic names assigned to values, acting as containers for storing data. They are dynamically typed in Python, meaning their type can change based on the assigned value.

Artificial Intelligence-Curriculum

2.4 Working with Data structure, Arrays, Vectors & Data Frames.

Data structures

- Data structures in Python (e.g., lists, tuples, dictionaries, and sets) are ways to store and organise data efficiently. They allow for easy access, modification, and management of data depending on the structure's properties.

Arrays

- Arrays (using libraries like numpy) and vectors are ordered collections of elements, typically of the same data type. Arrays support fast mathematical operations, while vectors are 1D arrays often used in linear algebra and machine learning.

Data Frame

- It is a two-dimensional, table-like data structure (from libraries like pandas) where data is stored in rows and columns. It's ideal for handling and manipulating structured data, similar to spreadsheets or SQL tables.

2.5 Syntax

- Rules and structure of code in programming.
- Defines correct keyword and symbol usage.
- Ensures code readability and functionality.
- Essential for error-free program execution.

2.6 Working with Numbers & Working with Strings

Working with Numbers

- Arithmetic operations like addition, subtraction, multiplication, and division.
- Handling numeric types like integers, floats, and complex numbers.

Working with Strings

- Manipulating text with functions like concatenation, slicing, and formatting.
- Supporting operations for string comparison, search, and transformation.

2.7 Conditional Statements

- Allow decision-making in programming based on conditions.
- Include if, else if, and else clauses.
- Enable branching logic for different outcomes.
- Support complex conditions with logical operators.

2.8 For Loop & While Loop

For Loop

- Iterates over a sequence or range of values.
- Commonly used for executing code a specific number of times.

While Loop

- Repeats code while a condition remains true.
- Useful for indeterminate iterations until a condition changes.

2.9 Lists, Tuples, Sets

Lists

- Ordered, mutable collections that can hold mixed data types; defined with [ ]. Supports indexing, slicing, and dynamic modifications.

Tuples

- Ordered, immutable collections that can hold mixed data types; defined with (). Ideal for fixed data that should not be altered.

Sets

- Unordered, mutable collections with unique elements; defined with { }. Used for eliminating duplicates and efficient membership testing.

Artificial Intelligence-Curriculum

2.10 Dictionaries & Functions

Dictionaries

- Stores data in key-value pairs.
- Allows fast lookup, insertion, and deletion by keys.

Functions

- Encapsulate reusable blocks of code.
- Can accept parameters and return values.

2.11 Pandas, NumPy, Matplotib packages.

Pandas

- Powerful library for data manipulation and analysis, Pandas provides data structures like DataFrames, allowing for easy handling, cleaning, and transformation of structured data.

NumPy

- A fundamental package for numerical computations, NumPy offers support for multi-dimensional arrays and a wide range of mathematical functions for operations on arrays and matrices.

Matplotib

- A popular plotting library used for creating static, interactive, and animated visualisations in Python, Matplotlib allows users to generate a wide variety of charts, including line plots, histograms, and scatter plots.

3.Intelligent Agents

3.1 Intelligent Agents

Autonomous systems that perceive their environment, make decisions, and take actions to achieve specific goals, often improving performance through learning and adaptation.

3.2 Rational Agents

These agents act to achieve the best possible outcome based on the information they have, making decisions using logic, knowledge, reasoning and aiming to act optimally in any given situation that maximises expected utility and

3.3 PEAS Representation

- Performance
- Environment
- Actuators
- Sensors

3.4 Types of AI Agents

- Simple reflex agents
- Model-based agent
- Goal-based agents
- Utility-based agents
- Learning agents

3.5 Uninformed Search Examples

Artificial Intelligence-Curriculum

4.Problem Solving

4.1 Search Algorithms

- Terminologies
- Transition Model
- Optimal Solution

4.2 Uninformed Search Algorithm

- Breadth First Search
- Depth First Search
- Depth Limited Search
- Uniform Cost Search
- Iterative Deepening Depth First Search
- Bidirectional Search

4.3 Informed (Heuristic) Search Algorithm

- Best First Search
- A* Search

4.4 Hill Climbing Algorithm

- No Backtracking
- State Space Diagram
- Simple Hill Climb
- Steepest Ascent Hill Climb
- Stochastic Hill Climb

5.Adversarial Search

5.1 Adversarial Search and Games

- Purpose: Game-playing
- Components: Players, States
- Deterministic Games
- Non Deterministic Games
- Zero Sum Game

5.2 Minimax Algorithm

- Goal: Optimal decision-making
- Type: Adversarial search
- Process: Evaluate moves, Minimise loss
- Components: Max player, Min player
- Use: Two-player games

5.3 Alpha-Beta Pruning

- Purpose: Optimise Minimax
- Function: Reduce search space
- Technique: Prune branches
- Efficiency: Faster evaluation
- Use: Game AI strategies

Artificial Intelligence-Curriculum

6.Machine Learning

6.1 Introduction to ML

6.2 Types of ML

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

6.3 Life Cycle of ML

- Gathering Data
- Data Preparation
- Data Wrangling
- Data Analysis
- Train Model
- Test Model
- Deployment

6.4 Supervised Learning

- Classification - Logistic Regression, Decision Trees, SVM, KNN, Naive Bayes,
- Regression - Linear Regression, Polynomial Regression,Ridge Regression,SVR

6.5 Unsupervised Learning

- Types: Clustering, Dimensionality Reduction, Association
- Techniques: K-Means, Hierarchical Clustering, PCA

6.6 Clustering Methods

- Partitioning Clustering
- Density Based Clustering
- Distribution Model Based Clustering
- Hierarchical Clustering

6.7 Association Rules

- Metrics of Association Rule Learning - Support, Lift, Confidence
- Types - Apriori Algorithm, Eclat Algorithm, F-P Growth Algorithm

7.Deep Learning

7.1 Introduction to Deep Learning

7.2 Architecture and Application

- Architecture - Deep Learning Network, Deep Belief Network
- Types - FFNN,CNN,Restricted Boltzmann Machine, Autoencoders

7.3 Deep Learning Algorithms

- Convolutional Neural Networks
- Long Short Term Memory Networks
- Recurrent Neural Networks
- Generative Adversarial Networks
- Radial Basis Function Networks

Artificial Intelligence-Curriculum

CAPSTONE PROJECTS

1 Text Classification With Tensorflow

- Mastery of TensorFlow Ecosystem: Gained practical experience in working with TensorFlow, TensorFlow Hub, and TensorFlow Datasets for loading, splitting, and processing data, especially for tasks like sentiment analysis.
- Transfer Learning for NLP: Learned to leverage pretrained models from TensorFlow Hub to efficiently implement transfer learning in natural language processing tasks, enhancing model performance with minimal effort.
- Model Building and Evaluation: Developed skills in building, training, and evaluating neural networks, focusing on performance metrics like accuracy and loss to fine-tune the model for better predictions.

2 Classification of Pet's Faces

- Image Data Handling and Label Encoding: The project effectively extracts pet breed names from image filenames, encodes them into numerical labels, and resizes images to a standard size (224x224), ensuring consistency for model training.
- Dataset Exploration and Visualization: The project uses Matplotlib to visualise the pet images and their distribution, helping to identify class imbalances and ensuring that the dataset is suitable for building a robust classifier
- TensorFlow Integration for Feature Extraction: It leverages TensorFlow's image processing capabilities to load, resize, and convert images into arrays, preparing the dataset for deep learning model development focused on pet breed classification.

3 Object Detection using Tensorflow

- Deep Learning Fundamentals: Gain a solid understanding of object detection algorithms, enhancing knowledge of convolutional neural networks (CNNs).
- TensorFlow Proficiency: Improve TensorFlow skills, learning to train and fine- tune models for accurate detection of multiple object classes.
- Data Annotation and Preprocessing: Learn the importance of data annotation and preprocessing to ensure reliable model training and enhance detection accuracy.

LIVE PROJECT

1 Landmark Detection

- Image Preprocessing and Data Management: Gained experience in loading, managing, and preprocessing large-scale image datasets using libraries like OpenCV, PIL, and pandas, essential for preparing data for deep learning models.
- Using VGG-19 for Transfer Learning: Applied the VGG-19 model, a powerful pre trained convolutional neural network, for transfer learning. This involved adapting the model to classify landmark images, showcasing the efficiency of using pre-trained architectures for specialised tasks with limited labelled data.
- End-to-End Workflow Integration: Developed skills in building an end-to- end workflow, integrating TensorFlow for model training, OpenCV for image handling, and Matplotlib for visualisation. This included fine-tuning the VGG- 19 model for improved performance on the landmark classification task.