**ANSWER NO:1**

**K-Nearest Neighbors (KNN):**

* **Strengths: Simple to understand and implement, doesn't make strong assumptions about the data distribution, effective for both classification and regression tasks.**
* **Weaknesses: Computationally expensive for large datasets, sensitive to irrelevant features, lacks interpretability, and may require tuning of the "k" parameter.**

**Support Vector Machines (SVM):**

* **Strengths: Effective in high-dimensional spaces, works well with small to medium-sized datasets, can handle both linear and non-linear classification problems using different kernels, offers good generalization, and has strong theoretical foundations.**
* **Weaknesses: Can be sensitive to the choice of kernel and its parameters, may not perform well with noisy or overlapping data, and can be computationally intensive for large datasets.**

**In summary, KNN is simple but computationally expensive, while SVM is powerful in high-dimensional spaces but requires careful kernel selection. Combining their strengths, such as using KNN as a local classifier within an SVM framework, can enhance performance in certain scenarios.**

**ANSWER NO:2   
Collect and preprocess documents: Gather the documents you want to rank and clean them by removing stopwords, punctuation, and stemming words if necessary.**

**Create a term-document matrix: Construct a matrix where rows represent documents and columns represent unique words (terms) from all the documents.**

**Calculate TF-IDF values: For each term in each document, compute its Term Frequency-Inverse Document Frequency (TF-IDF) value. TF measures the importance of a term in a document, while IDF measures its importance across all documents.**

**Normalize TF-IDF vectors: Normalize each document's TF-IDF vector to have a length of 1. This step ensures that document length doesn't bias the similarity calculation.**

**Choose a query document: Select the document you want to compare against the others. This will be your query document.**

**Calculate cosine similarity: Compute the cosine similarity between the TF-IDF vector of the query document and the TF-IDF vectors of all other documents. Cosine similarity measures the cosine of the angle between two vectors, with higher values indicating greater similarity.**

**Rank documents: Sort the documents based on their cosine similarity scores with the query document in descending order. The most similar documents will be at the top of the ranking.**

**Retrieve top-ranked documents: You now have a ranked list of documents based on their similarity to the query document. You can retrieve and present the top-ranked documents to the user as search results.**

**This process allows you to find and present documents that are most relevant to a given query by considering the importance of terms within documents and across the entire corpus using TF-IDF, and then measuring similarity using cosine similarity.**

**ANSWER NO:3**

* **Image Classification: Pre-trained models like VGG, ResNet, or Inception, originally trained on large image datasets**
* **Natural Language Processing (NLP): Models like BERT or GPT-3, pre-trained on vast text corpora, can be adapted for various NLP tasks such as sentiment analysis, text summarization, or language translation.**
* **Object Detection: Models like Faster R-CNN or YOLO, pre-trained on extensive datasets containing object annotations, can be used for object detection in custom applications, like tracking specific objects in security camera footage.**

**In each of these examples, the pre-trained models serve as a starting point, allowing developers to save time and resources by building upon the knowledge already embedded in the model.**

**ANSWER NO:4  
Transfer Learning in Deep Neural Networks:**

**Transfer learning involves using pre-trained neural network models as a starting point for solving a different but related task. Examples include:**

**Using a pre-trained image classification model (e.g., ResNet) for a new image classification task.**

**Fine-tuning a pre-trained language model (e.g., GPT-2) for text generation in a specific domain.**

**Using a pre-trained model for sentiment analysis and adapting it to classify product reviews.**

**Decision Trees vs. Random Forests:**

**Decision Trees:**

**Strengths:**

**Simple to understand and interpret.**

**Can handle both categorical and numerical data.**

**Prone to overfitting.**

**Weaknesses:**

**Tendency to overfit.**

**Limited predictive accuracy.**

**Random Forests (Ensemble of Decision Trees):**

**Strengths:**

**Reduces overfitting through averaging predictions from multiple trees.**

**Provides improved predictive accuracy and generalization.**

**Handles feature importance.**

**Weaknesses:**

**Can be computationally expensive.**

**May not perform well on very small datasets.**

**Preventing Overfitting in Artificial Neural Networks:**

**Use more training data.**

**Employ regularization techniques (L1, L2, dropout).**

**Simplify the architecture (reduce the number of layers or neurons).**

**Early stopping during training.**

**Cross-validation to evaluate model performance.**

**Batch normalization.**

**Data augmentation**

**ANSWER NO:5**

**Cross-validation to evaluate model performance.**

**Batch normalization.**

**Data augmentation**

**ANSWER NO:6**

**L1 vs. L2 Regularization in Linear Regression:**

**L1 Regularization (Lasso):**

**Adds the absolute values of coefficients to the cost function.**

**Encourages sparsity (some coefficients become exactly zero).**

**Can be used for feature selection.**

**L2 Regularization (Ridge):**

**Adds the squares of coefficients to the cost function.**

**Does not force coefficients to become exactly zero.**

**Distributes the impact of features more evenly.**

**Both L1 and L2 regularization help prevent overfitting by penalizing large coefficients.**

**ANSWER NO:7**

**Loss Functions in GANs:**

**Generator Loss: Typically, the generator minimizes the binary cross-entropy loss, encouraging it to produce realistic samples.**

**Discriminator Loss: The discriminator also minimizes binary cross-entropy loss, aiming to distinguish between real and generated samples.**

**Wasserstein GANs (WGANs) use the Wasserstein loss (Earth Mover's Distance) for more stable training.**

**ANSWER NO:8  
Supervised vs. Unsupervised vs. Reinforcement Learning:**

**Supervised Learning: Involves learning from labeled data, where the model is trained to make predictions based on input-output pairs (classification, regression).**

**Unsupervised Learning: Involves finding patterns or structure in unlabeled data (clustering, dimensionality reduction).**

**Reinforcement Learning: Focuses on training agents to make sequences of decisions in an environment to maximize a reward signal.**

**ANSWER NO:9  
Batch Gradient Descent vs. Stochastic Gradient Descent:**

**Batch Gradient Descent: Updates model parameters using the gradients computed over the entire training dataset. Slower but more stable convergence.**

**Stochastic Gradient Descent (SGD): Updates parameters using gradients computed for individual data points or small mini-batches. Faster but noisy convergence.**

**ANSWER NO:10  
Data Augmentation in Deep Learning: Data augmentation involves generating new training samples from existing data to increase diversity. Common techniques include:**

**Rotations, flips, and translations for images.**

**Adding noise or perturbations to data.**

**Text data: Synonym replacement, paraphrasing, or adding noise.**

**ANSWER NO:11  
Time Series Forecasting with ARIMA: ARIMA (AutoRegressive Integrated Moving Average) is a model for time series forecasting. It involves differencing the data until it becomes stationary, fitting auto-regressive and moving average terms, and making predictions. Suitable for stationary, univariate time series data.**

**ANSWER NO:12**

**GAN Architecture:**

**Generator: Creates fake data samples.**

**Discriminator: Tries to distinguish between real and fake data.**

**Training: Generator tries to produce data that can fool the discriminator, and the discriminator aims to correctly classify real vs. fake data. This adversarial process continues until the generator produces realistic data.**

**ANSWER NO:13**

**LSTMs in RNNs and Vanishing Gradient: LSTMs are a type of RNN designed to address the vanishing gradient problem. They use gates to control the flow of information, allowing them to capture long-range dependencies in sequences. The forget gate, input gate, and output gate control information flow through time, making them more effective for long sequences.**

**ANSWER NO:14**

**Q14 no: Data Preprocessing in NLP: Common steps:**

**Tokenization: Split text into words or sub word units.**

**Lowercasing: Convert text to lowercase.**

**Stop word Removal: Remove common, uninformative words.**

**Lemmatization or stemming: Reduce words to their base or root form.**

**Vectorization: Convert text to numerical form (e.g., TF-IDF or word embeddings).**

**ANSWER NO:15**

**Handling Missing Data:**

**Mean Imputation: Replace missing values with the mean of the feature.**

**Predictive Modeling: Build a model to predict missing values based on other features.**

**Deletion: Remove rows or features with missing data (not ideal if data is valuable).**

**Impute with median/mode for non-normally distributed data.**