Code : 20CS511

Reg.No. | | | | | | | | | | |

MODEL ANSWERS

V Semester Diploma Make-up Examination, Sep-2023

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Duration: 3 Hours **Max.Marks:** 100

Instructions: Answer one full question from each Section.

SECTION-I

1.a Describe AI and its applications in various fields. 10 Marks

Artificial Intelligence (AI) is a branch of computer science that aims to create intelligent machines capable of performing tasks that typically require human intelligence. AI systems can learn from data, adapt to new information, and make decisions or solve problems. Here are 10 applications of AI in various fields:

i. **Healthcare:**

- **Medical Diagnosis:** AI algorithms can analyze medical images, such as X-rays and MRIs, to assist doctors in diagnosing diseases like cancer.
- **Drug Discovery:** AI accelerates drug discovery by predicting potential drug candidates and their efficacy.

ii. **Finance:**

- **Algorithmic Trading:** AI-driven algorithms analyze financial data and execute trades at high speeds, optimizing investment portfolios.
- **Fraud Detection:** AI can identify fraudulent transactions by detecting unusual patterns and behaviors in financial data.

iii. **Autonomous Vehicles:**

- **Self-Driving Cars:** AI enables cars to perceive their environment, make real-time decisions, and navigate autonomously.

iv. **Natural Language Processing (NLP):**

- **Chatbots:** AI-powered chatbots engage in natural language conversations with users to provide customer support or answer questions.
- **Language Translation:** NLP models like Google Translate can translate text and speech between languages.

v. **Education:**

- **Personalized Learning:** AI systems adapt educational content and assessments to individual student needs.
- **Automated Grading:** AI can automatically grade assignments and provide instant feedback to students.

vi. **E-commerce:**

- **Recommendation Systems:** AI algorithms analyze user behavior and preferences to recommend products and content.
- **Inventory Management:** AI optimizes inventory levels and supply chain logistics.

vii. **Manufacturing:**

- **Quality Control:** AI-powered robots and cameras inspect products for defects in real-time.
- **Predictive Maintenance:** AI predicts equipment failures and schedules maintenance to minimize downtime.

viii. **Robotics:**

- **Industrial Robots:** AI-driven robots perform tasks like welding, assembly, and pick-and-place operations in manufacturing.
- **Social Robots:** AI-powered robots interact with humans in settings like healthcare, customer service, and companionship.

ix. **Agriculture:**

- **Precision Agriculture:** AI analyzes data from sensors, drones, and satellites to optimize crop management, irrigation, and yield prediction.

x. **Entertainment:**

- **Content Creation:** AI can generate music, art, and even write articles or scripts.
- **Gaming:** AI enhances video game experiences through intelligent opponents and dynamic storytelling.

1.b How AI Software Development life cycle differs from traditional software development. Explain. 05 Marks

Aspect	AI SDLC	Traditional SDLC
1. Data-Centric Approach	Emphasizes data collection, cleaning, and prep.	Data supports functionality but is secondary.
2. Iterative Model Training	Involves iterative model training and tuning.	Typically follows a more linear or phased path.
3. Algorithm Development	Focuses on complex ML algorithms and models.	Emphasizes deterministic, rule-based algorithms.
4. Data Quantity and Quality	Requires large, high-quality datasets.	Data quality and quantity may vary by project.
5. Interpretability and Explainability	Needs to ensure models are interpretable.	Less focus on interpretability in most cases.

1.c Summarize the challenges associated with Machine Learning. 05 Marks

1. **Data Quality and Quantity:**
   - Machine learning models require large and high-quality datasets for training. Obtaining clean and representative data can be challenging, and insufficient data can lead to model underfitting.
2. **Overfitting and Underfitting:**
   - Finding the right balance in model complexity is tricky. Overly complex models may overfit (perform well on training data but poorly on new data), while overly simple models may underfit (fail to capture underlying patterns).
3. **Interpretability and Explainability:**

- Many machine learning models, especially deep learning models, are complex and hard to interpret. Explaining why a model made a particular prediction is challenging, which can be problematic in sensitive domains.

4. **Bias and Fairness:**
    - Machine learning models can inherit biases present in the training data, leading to unfair or discriminatory outcomes. Ensuring fairness and mitigating bias is a significant challenge.
5. **Model Deployment and Maintenance:**
    - Deploying machine learning models into production environments and maintaining them can be complex. Issues related to model drift, scalability, and version control need to be addressed.

2.a Perform the following operations/write code snippet on Car manufacturing company dataset "auto-mpg.csv" given below using pandas. 10 Marks

i) Read data from a file.

ii) Calculate mean value of "horsepower".

iii) Calculate Standard Deviation value of "acceleration".

iv) Get the number of cars manufactured in each year.

mpg	cylinders	displacement	horsepower	weight	acceleration	model year	car name
18	8	307	130	3504	12	71	Chevrolet
15	8	350	165	3693	11.5	70	Skylark
18	8	318	150	3436	11	72	Plymouth
17	8	302	140	3449	10.5	70	Ford
14	8	455	225	4425	10	71	Pontiac
15	8	390	190	3850	8.5	70	Ambassador

i) Read data from a file.

```
df = pd.read_csv('auto-mpg.csv')
print(df.head())
```

ii) Calculate mean value of "horsepower".

```
horsepower_mean = df['horsepower'].mean()
print("\nMean horsepower:", horsepower_mean)
```

iii) Calculate Standard Deviation value of "acceleration".

```
acceleration_std = df['acceleration'].std()
print("\nStandard Deviation of acceleration:", acceleration_std)
```

iv) Get the number of cars manufactured in each year.

```
manufactured_by_year = df['model_year'].value_counts().sort_index()
print("\nNumber of cars manufactured in each year:")
```

print(manufactured_by_year)

2.b Explain how is AI software development life cycle different from traditional software development?

10 Marks

Aspect	AI SDLC	Traditional SDLC
1. Data-Centric Approach	Focuses on data collection, cleaning, and preprocessing.	Data is used to support functionality but is not central.
2. Iterative Model Training	Involves iterative training and tuning of machine learning models.	Typically follows a more linear or phased approach.
3. Algorithm Development	Requires the creation of complex machine learning algorithms and models.	Emphasizes deterministic, rule-based algorithms.
4. Data Quantity and Quality	Demands large, high-quality datasets for effective model training.	Data quality and quantity may vary by project.
5. Interpretability and Explainability	Emphasizes the need to ensure models are interpretable and explainable.	Less focus on interpretability in most cases.
6. Continuous Learning	Requires staying updated with the latest AI research and model advancements.	Less focus on continuous learning once the software is deployed.
7. Ethical Considerations	Involves ethical considerations, especially regarding data privacy and bias.	Ethics are important but may not be as data-centric.
8. Uncertainty and Experimentation	Acknowledges uncertainty in model outcomes; experimentation is common.	Aims for predictable outcomes based on requirements.
9. Specialized Skillset	Demands expertise in machine learning, data science, and domain knowledge.	Relies on software engineering skills and domain expertise.
10. Deployment and Monitoring	Requires continuous monitoring of deployed models for performance and updates.	Focuses on releasing stable software versions with occasional updates.

SECTION-II

3.a Handling missing values in a dataset is a crucial data pre-processing step, as missing data can lead to biased or incorrect results in your analysis or machine learning models. Elaborate on how missing values in the data sets can be handled. **10 Marks**

There are several ways to handle missing values in a dataset, including:

i. Dropping the rows or columns that contain missing values: This is a simple method, but it can lead to loss of information if the percentage of missing values is high.

Example: df=train_df.drop(['Dependents'],axis=1) df.isnull().sum()

ii. Imputing the missing values: This method involves replacing the missing values with a substitute value, such as the mean or median of the non-missing values. This method can be useful if the percentage of missing values is low, but it can lead to biased results if the missing data is not missing at random.

Example : df =train_df.dropna(axis=0) df.isnull().sum()

iii. Using machine learning algorithms: Some machine learning algorithms can handle missing values automatically and make predictions based on the available data. For example, decision trees and random forests can handle missing values and split the data based on the available features.

iv. **Using advanced imputation technique** like multiple imputation, this method creates multiple imputed datasets and then combine them using some statistical methods.

• Replacing with Arbitrary Value

If you can make an educated guess about the missing value, then you can replace it with some arbitrary value using the following code.

Ex: In the following code,we are replacing the missing values of the 'Dependents'column with'0'
train_df['Dependents']=train_df['Dependents'].fillna(0)

• Replacing with Mean

This is the most common method of imputing missing values of numeric columns.One can use the 'fillna' method for imputing the columns 'Loan Amount' with theme an oft he respective column values as below

train_df['LoanAmount'].fillna(train_df['LoanAmount'].mean())

• Replacing with Mode

Mode is the most frequently occurring value. It is used in the case of categorical features. You can use the 'fillna' method for imputing the categorical columns 'Gender', 'Married', and'Self_Employed'.

train_df['Gender'].fillna(train_df['Gender'].mode()[0])

• Replacing with Median

Median is the middlemost value. It's better to use the median value for imputation in the case

ofoutliers. You can use 'fillna' method for imputing the column 'Loan_Amt' with the median value.
train_df['Loan_Amt']=train_df['Loan_Amt'].fillna(train_df['Loan_Amt'].median()[0])

v. Keep the missing value as is

Sometimes missing data is very less number of rows (say less than 3%) then we can simply ignore the missing data.There is no hard rule to keep the missing data, it depends on us. Remove data objects with missing values (Deleting the entire column) 10 Marks

3.b A dataset is given to you for creating machine learning model. What are the steps followed before using the data for training the model? Elaborate each step. 10 Marks

Data Exploration: The first step is to explore the data and understand the characteristics of the dataset. This includes understanding the number of observations and variables, the data types of each variable, and the distribution of the data. This can be done by using summary statistics and visualizations such as histograms, box plots, and scatter plots.

Data Cleaning: The next step is to clean the data. This includes handling missing or corrupted data, removing outliers, and addressing any other data quality issues. This step is important because dirty data can lead to inaccurate or unreliable models.

Data Transformation: After cleaning the data, it may be necessary to transform the data to make it suitable for the machine learning model. This can include normalizing the data, scaling the data, or creating new variables.

Feature Selection: Once the data is cleaned and transformed, it is important to select the relevant features that will be used to train the model. This step can be done by using techniques such as correlation analysis, principal component analysis, or mutual information.

Data Splitting: The next step is to split the data into training, validation, and test sets. The training set is used to train the model, the validation set is used to tune the model's parameters, and the test set is used to evaluate the model's performance.

Feature Engineering: This step is to create new features that will be useful in the model. This can include creating interaction terms, polynomial terms, or binning variables.

Evaluation Metric: Selecting the right evaluation metric will help to evaluate the model's performance. Common evaluation metrics include accuracy, precision, recall, F1 score, and area under the ROC curve.

Model Selection: After the data is prepared, the next step is to select the appropriate machine learning model. This can be done by comparing the performance of different models using the evaluation metric.

Model Training: Once the model is selected, it is trained using the training dataset.

Model Evaluation: Finally, the model's performance is evaluated using the test dataset and the evaluation metric selected.

4.a Create two series as shown using pd. series() function. **10 Marks**

Series A = [20, 30, 40, 50, 60]

Series B = [50, 60, 70, 80, 90]

(i) Get the items not common to both.

(ii) Identify the smallest and largest element in the Series A.

(iii) Find the sum of Series B.

(iv) Calculate mean in the Series A.

(v) Find median in the given Series B.

Create Series A and Series B

series_a = pd.Series([20, 30, 40, 50, 60])

series_b = pd.Series([50, 60, 70, 80, 90])

(i) Get the items not common to both

not_common = series_a[~series_a.isin(series_b)].append(series_b[~series_b.isin(series_a)])

(ii) Identify the smallest and largest element in Series A

smallest_a = series_a.min()

largest_a = series_a.max()

(iii) Find the sum of Series B

sum_b = series_b.sum()

(iv) Calculate the mean in Series A

mean_a = series_a.mean()

(v) Find the median in Series B

median_b = series_b.median()

4.b Referring to the number of variables or features in a dataset and the focus of analysis. Explain univariate & multivariate data types with examples. 10 Marks

Univariate and multivariate data types refer to the number of variables or features in a dataset and the focus of analysis. These terms are fundamental in statistics and data analysis. Let's explore each type and provide examples:

1. Univariate Data: Univariate data analysis deals with a single variable or feature in a dataset. The primary objective is to understand the distribution and characteristics of that individual variable. Univariate analysis is typically used when you want to explore or summarize one aspect of the data.

Example - Univariate Analysis: Suppose you have a dataset containing the ages of a group of people. You are interested in understanding the distribution of ages in this dataset. You perform univariate analysis on the "Age" variable, which may involve creating histograms, calculating summary statistics like mean and median, and visualizing the data using box plots.

2. Multivariate Data: Multivariate data analysis involves the analysis of two or more variables simultaneously to understand the relationships and patterns that may exist among them. Multivariate analysis is used when you want to explore how variables interact with each other or when you want to predict one variable based on others. It is common in statistical modeling and machine learning.

Example - Multivariate Analysis: Consider a dataset containing information about houses, including "Square Footage" and "Number of Bedrooms." You are interested in predicting the "Price" of a house based on both of these variables. This scenario involves multivariate analysis, as you are considering the interaction between two variables to predict a third.

SECTION – III

5.a A Machine learning model was built to classify spam emails as "spam"(1) or "not spam"(0). The confusion matrix for the model is as shown below. Evaluate accuracy, precision, recall, specificity and F1-Score. 10 Marks

		Actual	
		1	0
Predicted	1	**140**	**10**
	0	**5**	**50**

Accuracy:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{140+50}{140+50+10+5} = \frac{190}{205} \approx 0.9268$$

Precision:

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{140}{140+10} = \frac{140}{150} = 0.9333$$

Recall:

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{140}{140+5} = \frac{140}{145} \approx 0.9655$$

Specificity:

$$\text{Specificity} = \frac{TN}{TN+FP} = \frac{50}{50+10} = \frac{50}{60} \approx 0.8333$$

F1-Score:

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}} = \frac{2 \cdot 0.9333 \cdot 0.9655}{0.9333+0.9655} \approx 0.9491$$

5.b Explain Supervised and Unsupervised learning with examples. 5 Marks

Supervised learning is a type of machine learning where the algorithm is trained on a labeled dataset, meaning that the input data is paired with corresponding output labels or target values. The goal of supervised learning is to learn a mapping or relationship between the input features and the target labels, so that it can make predictions on new, unseen data.

Example of Supervised Learning:

Let's consider a classic example of supervised learning: email spam classification. In this scenario, you have a dataset of emails, each labeled as either "spam" or "not spam" (also called "ham"). The input features might include various characteristics of the emails, such as the sender's address, subject line, and the content of the email. The target labels are "spam" or "not spam."

Unsupervised Learning:

Unsupervised learning, on the other hand, deals with unlabeled data, where the algorithm tries to find patterns, structures, or groupings in the data without the presence of explicit target labels. It is often used for exploratory data analysis and data clustering.

Example of Unsupervised Learning:

A common example of unsupervised learning is clustering. Consider a dataset of customer purchasing behavior in a retail store, where each data point represents a customer's purchase history. In this case, you don't have labels indicating specific customer segments; the goal is to group similar customers together based on their buying patterns.

5.c Compare overfitting with under-fitting. 5 Marks

Aspect	Overfitting	Underfitting
Model Complexity	High model complexity, often overly complex.	Low model complexity, often too simple.
Training Performance	Excellent on the training data (low training error).	Poor performance on the training data (high training error).
Generalization	Poor generalization to new, unseen data.	Poor generalization to both training and test data.
Characteristics	Captures noise and fluctuations in the training data.	Fails to capture underlying patterns and trends.
Error Trends	Training error decreases, but test error increases.	Both training and test errors are high.
Bias-Variance Trade-off	High variance (model is too sensitive to data).	High bias (model is too simple to capture patterns).
Remedies	Reduce model complexity, use regularization.	Increase model complexity, gather more data.

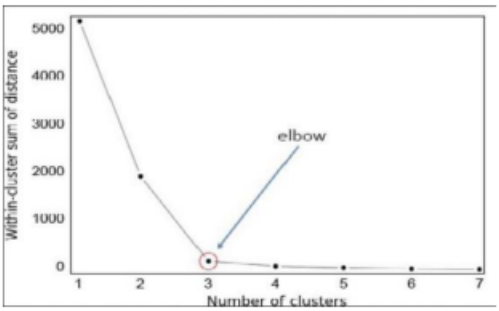
6.a How to Choose the Right Number of Clusters in k-means clustering? Explain any one method. 5 Marks

Three approaches to find the optimal number of clusters:

- The elbow method
- The optimization of the silhouette coefficient
- The gap statistic

Elbow method

1. Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
2. For each k, calculate the total within-cluster sum of square (wss).
3. Plot the curve of wss according to the number of clusters k.
4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.



Average silhouette method

The algorithm is similar to the elbow method and can be computed as follow:

1. Compute clustering algorithm (e.g., k-means clustering) for different values of k. For instance, by varying k from 1 to 10 clusters.
2. For each k, calculate the average silhouette of observations (avg.sil).
3. Plot the curve of avg.sil according to the number of clusters k.
4. The location of the maximum is considered as the appropriate number of clusters.

Gap statistic method

The algorithm works as follow:

1. Cluster the observed data, varying the number of clusters from k = 1, …, kmax, and compute the corresponding total within intra-cluster variation W_k.

2. Generate B reference data sets with a random uniform distribution. Cluster each of these reference data sets with varying number of clusters k = 1, …, kmax, and compute the corresponding total within intra-cluster variation W_{kb}.

3. Compute the estimated gap statistic as the deviation of the observed Wk value from its expected value Wkb under the null hypothesis: Gap(k)=1B∑b=1Blog(W∗kb)−log(Wk)

4. Compute also the standard deviation of the statistics.

5. Choose the number of clusters as the smallest value of k such that the gap statistic is within one standard deviation of the gap at k+1: Gap(k)≥Gap(k + 1)−sk + 1.

6.b Compare "Classification algorithms" with "Clustering algorithm". 5 Marks

Aspect	Classification Algorithms	Clustering Algorithms
Objective	Assign data points to predefined classes or labels based on input features.	Group data points based on similarity or patterns, discovering clusters.
Supervision	Supervised learning: Requires labeled training data for model training.	Unsupervised learning: Works with unlabeled data; no predefined classes.
Output	Assigns each data point to a specific class or label with a clear meaning.	Groups data points into clusters; clusters may lack predefined meanings.
Evaluation Metrics	Accuracy, precision, recall, F1-score, ROC AUC, confusion matrix, etc.	Silhouette score, Davies-Bouldin index, Dunn index, etc., for cluster quality.
Examples of Algorithms	Logistic Regression, Decision Trees, Support Vector Machines, etc.	K-Means, Hierarchical Clustering, DBSCAN, Gaussian Mixture Models, etc.

6.c Explain with examples: Scalars, Vectors, Matrices, Tensors and Gradients in Linear Algebra. 10 Marks

1. **Scalars:**
    - A scalar is a single numerical value, typically a real number.
    - Scalars have magnitude but no direction.
    - Examples: Temperature (e.g., 25°C), mass (e.g., 5 kg), and speed (e.g., 60 km/h).
2. **Vectors:**
    - A vector is an ordered collection of scalars, often represented as an array.
    - Vectors have both magnitude and direction and are commonly used to represent quantities like displacement or force.
    - Examples:
        - Position vector in 2D: $\mathbf{v}=[x,y]$
        - Velocity vector: $\mathbf{v}=[3,4]$ m/s (3 m/s in the x-direction and 4 m/s in the y-direction).
3. **Matrices:**
    - A matrix is a rectangular array of numbers, symbols, or expressions arranged in rows and columns.
    - Matrices are used to represent linear transformations and to solve systems of linear equations.
    - Examples:
        - Identity Matrix (3×3): $I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
        - Transformation matrix: $T = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$ (scales by 2 in x-direction and 3 in y-direction).
4. **Tensors:**
    - Tensors generalize scalars, vectors, and matrices to higher dimensions.

- They are multi-dimensional arrays with a uniform type (e.g., all numbers).
- Examples:
    - A color image can be represented as a 3D tensor (width x height x color channels).
    - A video sequence can be represented as a 4D tensor (frames x width x height x color channels).

5. **Gradients:**
    - A gradient is a vector that represents the rate of change of a scalar function at each point in space.
    - It points in the direction of the steepest increase of the function and has a magnitude equal to the slope.
    - Gradients are used in optimization and machine learning for finding optimal solutions.
    - Example: In machine learning, when training a neural network, the gradient of the loss function with respect to the model's parameters (weights and biases) is computed and used to update the parameters during optimization (e.g., gradient descent).

SECTION – IV

7.a N-grams are a type of linguistic model used in natural language processing (NLP) and computational linguistics. Consider the given sentence: 10 Marks

"Artificial Intelligence is a branch of computer science that focuses on creating intelligent machines capable of performing tasks that typically require human intelligence."

i. Generate bi grams for the above sentence

ii. Generate tri-grams for the above sentence

Bi-grams are pairs of consecutive words in a sentence.

1. "Artificial Intelligence"
2. "Intelligence is"
3. "is a"
4. "a branch"
5. "branch of"
6. "of computer"
7. "computer science"
8. "science that"
9. "that focuses"
10. "focuses on"
11. "on creating"
12. "creating intelligent"
13. "intelligent machines"
14. "machines capable"
15. "capable of"

16. "of performing"
17. "performing tasks"
18. "tasks that"
19. "that typically"
20. "typically require"
21. "require human"
22. "human intelligence"

ii. **Tri-grams:**

Tri-grams are groups of three consecutive words in a sentence.

1. "Artificial Intelligence is"
2. "Intelligence is a"
3. "is a branch"
4. "a branch of"
5. "branch of computer"
6. "of computer science"
7. "computer science that"
8. "science that focuses"
9. "that focuses on"
10. "focuses on creating"
11. "on creating intelligent"
12. "creating intelligent machines"
13. "intelligent machines capable"
14. "machines capable of"
15. "capable of performing"
16. "of performing tasks"
17. "performing tasks that"
18. "tasks that typically"
19. "that typically require"
20. "typically require human"
21. "require human intelligence"

7.b Explain how data exploration, pre-processing of data and splitting of data are performed on datasets.

10 Marks

Data exploration, data pre-processing, and data splitting are essential steps in the data preprocessing pipeline of a machine learning.

1. **Data Exploration:**

Data exploration is the initial phase of understanding and getting insights from your dataset. Its purpose is to gain familiarity with the data, identify patterns, and check for potential issues. Common tasks in data exploration include:

- **Summary Statistics:** Compute basic statistics such as mean, median, standard deviation, and percentiles for numerical features.
- **Data Visualization:** Create plots and charts (histograms, scatter plots, box plots, etc.) to visualize the distribution of data, relationships between variables, and potential outliers.
- **Handling Missing Data:** Identify and decide how to handle missing values in the dataset, whether by imputation or removal.
- **Feature Exploration:** Examine the characteristics of each feature, such as its data type, uniqueness, and relevance to the problem.
- **Correlation Analysis:** Investigate the correlation between features to understand how they relate to each other.
- **Outlier Detection:** Identify and address outliers that might affect model performance.

Data exploration helps you make informed decisions about data cleaning, feature selection, and model choice.

2. **Data Pre-processing:**

Data pre-processing is the stage where you prepare the data for model training. It involves cleaning and transforming the dataset to make it suitable for machine learning algorithms. Common data pre-processing tasks include:

- **Data Cleaning:** Address missing values, duplicate records, and inconsistencies in the data.
- **Feature Scaling:** Normalize or standardize numerical features to bring them to a similar scale, which can improve the performance of some algorithms.
- **Categorical Encoding:** Convert categorical variables into numerical representations (e.g., one-hot encoding or label encoding).
- **Feature Engineering:** Create new features or transform existing ones to capture relevant information for the problem.
- **Dimensionality Reduction:** Reduce the number of features, if necessary, using techniques like PCA (Principal Component Analysis).
- **Data Splitting:** Divide the data into training, validation, and test sets for model training, tuning, and evaluation.

Data pre-processing ensures that your dataset is in a suitable format for training and testing machine learning models.

3. **Splitting Data:**

Splitting the dataset into multiple subsets is crucial to assess the model's performance accurately. The common data splits include:

- **Training Set:** The largest portion of the dataset used to train the machine learning model. It helps the model learn patterns and relationships from the data.
- **Validation Set:** A smaller subset used for hyper-parameter tuning and model selection. It helps prevent overfitting by providing an independent evaluation.
- **Test Set:** Another separate subset used for the final evaluation of the model's performance. It gives an estimate of how the model will perform on unseen data.

The typical split ratios are 70-80% for the training set, 10-15% for the validation set, and 10-15% for the test set.

8.a With examples demonstrate Stemming and Lemmatization normalization techniques. 10 Marks

Stemming and lemmatization are text normalization techniques used in natural language processing (NLP) to reduce words to their base or root forms. Let's demonstrate both techniques with examples:

Stemming:

Stemming reduces words to their root or base form by removing prefixes or suffixes. It's a more aggressive technique compared to lemmatization and may result in non-real words.

Example using Python's NLTK library for stemming:

```
from nltk.stem import PorterStemmer

stemmer = PorterStemmer()

words = ["jumping", "jumps", "jumped", "running", "runner"]

stemmed_words = [stemmer.stem(word) for word in words]

print(stemmed_words)
```

Output:

```
['jump', 'jump', 'jump', 'run', 'runner']
```

In this example, you can see that stemming has reduced the words to their root forms, but it might not always result in valid words.

Lemmatization:

Lemmatization, on the other hand, reduces words to their base or dictionary form (lemma) while considering the word's part of speech. It results in valid words.

Example using Python's NLTK library for lemmatization:

```
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

words = ["jumping", "jumps", "jumped", "running", "runner"]

lemmatized_words = [lemmatizer.lemmatize(word, pos='v') for word in words]

print(lemmatized_words)
```

Output:

```
['jump', 'jump', 'jump', 'run', 'runner']
```

OR

Stemming:

Stemming is a text normalization technique that reduces words to their base or root forms by removing prefixes or suffixes. It aims to achieve this reduction by chopping off parts of the word.

- Original Word: "Jumping"
- Stemmed Word: "Jump"

In this example, the suffix "ing" was removed to obtain the root form "jump."

Lemmatization:

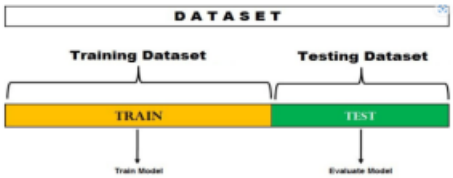
Lemmatization is another text normalization technique that reduces words to their base or dictionary forms while considering the word's part of speech. It results in valid words.

- Original Word: "Jumping"
- Lemmatized Word: "Jump"

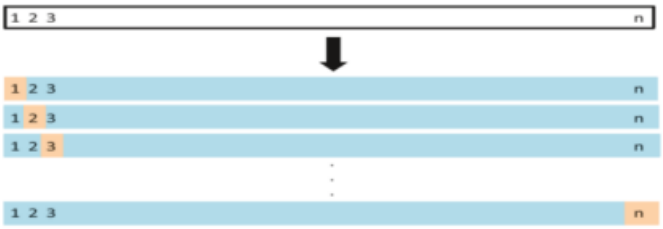
In this example, lemmatization considers the part of speech and converts "jumping" to the base form "jump," ensuring that the resulting word is valid and found in the dictionary.

8b. Explain any 2 techniques of cross validation used in Machine Learning. 5 Marks

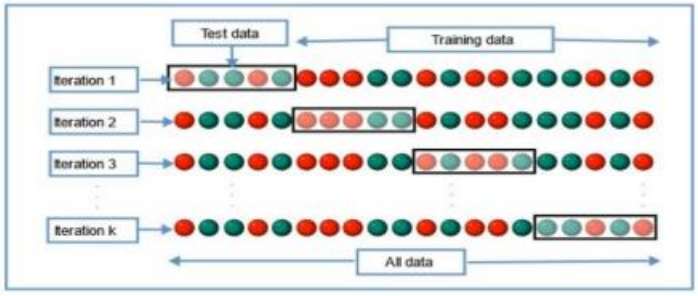
1. **Hold Out method**
   This is the simplest evaluation method and is widely used in Machine Learning projects. Here the entire dataset(population) is divided into 2 sets – train set and test set. The data can be divided into 70-30 or 60- 40, 75-25 or 80-20, or even 50-50 depending on the use case. As a rule, the proportion of training data has to be larger than the test data.



2. **Leave One Out Cross-Validation**
   In this method, we divide the data into train and test sets – but with a twist. Instead of dividing the data into 2 subsets, we select a single observation as test data, and everything else is labelled as training data and the model is trained. Now the 2nd observation is selected as test data and the model is trained on the remaining data.



3. **K-Fold Cross-Validation**
   In this resampling technique, the whole data is divided into k sets of almost equal sizes. The first set is selected as the test set and the model is trained on the remaining k-1 sets. The test error rate is then calculated after fitting the model to the test data.In the second iteration, the 2nd set is selected as a test set 17 and the remaining k-1 sets are used to train the data and the error is calculated. This process continues for all the k sets.



8c. Brief explain different stages involved in the Machine Learning Operations (MLOps) lifecycle. 5 Marks

MLOps stands for Machine Learning Operations. MLOps is focused on streamlining the process of deploying machine learning models to production, and then maintaining and monitoring them. MLOps is a collaborative function, often consisting of data scientists, ML engineers, and DevOps engineers. The word MLOps is a compound of two different fields i.e. machine learning and DevOps from software engineering.

1. ML Development: This is the basic step that involves creating a complete pipeline beginning from data processing to model training and evaluation codes.

2. Model Training: Once the setup is ready, the next logical step is to train the model. Here, continuous training functionality is also needed to adapt to new data or address specific changes.

3. Model Evaluation: Performing inference over the trained model and checking the accuracy/correctness of the output results.

4. Model Deployment: When the proof of concept stage is accomplished, the other part is to deploy the model according to the industry requirements to face the real-life data.

5. Prediction Serving: After deployment, the model is now ready to serve predictions over the incoming data.

6. Model Monitoring: Over time, problems such as concept drift can make the results inaccurate hence continuous monitoring of the model is essential to ensure proper functioning.

7. Data and Model Management: It is a part of the central system that manages the data and models. It includes maintaining storage, keeping track of different versions, ease of accessibility, security, and configuration across various cross-functional teams.

Section - V

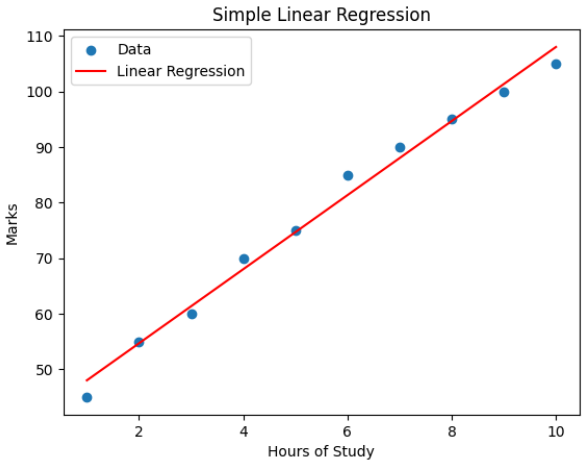
9.a Demonstrate simple linear regression considering a dataset that has two variables: 10 Marks

"Marks" (dependent variable)

"**Hours of study"** (independent variable)

```
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Sample data (hours of study and corresponding marks)
hours_of_study = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
marks = np.array([45, 55, 60, 70, 75, 85, 90, 95, 100, 105])

# Reshape the data (required for sklearn)
hours_of_study = hours_of_study.reshape(-1, 1)

# Create a linear regression model
model = LinearRegression()

# Fit the model to the data
model.fit(hours_of_study, marks)

# Predict marks for a new value of hours of study
new_hours_of_study = np.array([[11]])
predicted_marks = model.predict(new_hours_of_study)
```

```
# Get the slope (coefficient) and intercept of the regression line
slope = model.coef_[0]
intercept = model.intercept_

# Plot the data points and regression line
plt.scatter(hours_of_study, marks, label='Data')
plt.plot(hours_of_study, model.predict(hours_of_study), color='red', label='Linear Regression')
plt.xlabel('Hours of Study')
plt.ylabel('Marks')
plt.title('Simple Linear Regression')
plt.legend()
plt.show()

# Display the regression equation
print(f"Regression Equation: Marks = {slope:.2f} * Hours of Study + {intercept:.2f}")

# Predicted marks for the new value of hours of study
print(f"Predicted Marks for 11 hours of study: {predicted_marks[0]:.2f}")
```



OR

In simple linear regression, we aim to model the relationship between two variables: a dependent variable (in this case, "Marks") and an independent variable (in this case, "Hours of study"). The goal is to find a linear equation that best represents how "Marks" varies with changes in "Hours of study."

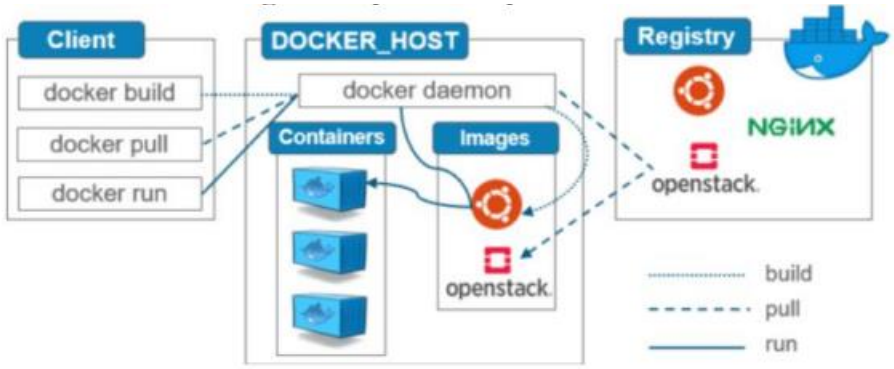
Here are the key components of simple linear regression:

1. **Dependent Variable ("Marks"):** This is the variable we want to predict or explain. In this example, "Marks" is the dependent variable, and we want to understand how it is influenced by "Hours of study."
2. **Independent Variable ("Hours of study"):** This is the variable that is believed to influence or explain changes in the dependent variable. In this case, "Hours of study" is the independent variable, and we believe that the more hours a student studies, the higher their "Marks" will be.

3. **Scatter Plot:** Before applying regression analysis, it's common to create a scatter plot. Each point on the plot represents a data point with "Hours of study" on the x-axis and "Marks" on the y-axis. The scatter plot helps us visualize the relationship between the two variables.
4. **Regression Line:** The objective of simple linear regression is to find the best-fitting line (the regression line) that represents the relationship between the variables. This line is characterized by two parameters: the slope (m) and the intercept (b).
    - **Slope (m):** The slope of the regression line represents how much "Marks" is expected to change for a one-unit change in "Hours of study." A positive slope indicates a positive relationship (more study hours lead to higher marks), while a negative slope indicates a negative relationship.
    - **Intercept (b):** The intercept is the point where the regression line intersects the y-axis. It represents the predicted value of "Marks" when "Hours of study" is zero. In most cases, this intercept might not have a practical interpretation in the context of the problem.
5. **Regression Equation:** The regression equation expresses the relationship between the variables. For simple linear regression, the equation is:

    Marks = (Slope * Hours of study) + Intercept
6. **Model Evaluation:** After fitting the regression line to the data, you can evaluate the model's goodness of fit using metrics like the coefficient of determination (R-squared), mean squared error (MSE), or other relevant metrics.

9.b With a neat diagram explain components of Docker. 10 Marks



Components of Docker

1. Docker Client Docker client uses commands and REST APIs to communicate with the Docker Daemon (Server). When a client runs any Docker command on the Docker client terminal, the client terminal sends these Docker commands to the Docker daemon. Docker daemon receives these commands from the Docker client in the form of command and REST API's request.
   Docker Client uses Command Line Interface (CLI) to run the following commands –
   • Docker build
   • Docker pull
   • Docker run
2. Docker Registry
   Docker Registry manages and stores the Docker images.There are two types of registries in the Docker Pubic Registry - Public Registry is also called as Docker hub. Private Registry - It is used to share images within the enterprise.
3. Docker Daemon
   This is the background process that runs on the host machine and manages the containers. It is responsible for creating, starting, stopping, and removing containers, as well as managing their network and storage resources.
4. Docker Images

Docker images are the read-only binary templates used to create Docker Containers. It uses a private container registry to share container images within the enterprise and also uses public container registry to share container images within the whole world. Metadata is also used by docket images to describe the container's abilities.

5. Docker Containers
   Containers are the structural units of Docker, which is used to hold the entire package that is needed to run the application. The advantage of containers is that it requires very less resources. In other words, we can say that the image is a template, and the container is a copy of that template.

10.a Summarize any two cloud deployment models . 10 Marks

1.Public Cloud

The public cloud makes it possible for anybody to access systems and services. The public cloud may be less secure as it is open to everyone. The public cloud is one in which cloud infrastructure services are provided over the internet to the general people or major industry groups. The infrastructure in this cloud model is owned by the entity that delivers the cloud services, not by the consumer. It is a type of cloud hosting that allows customers and users to easily access systems and services. This form of cloud computing is an excellent example of cloud hosting, in which service providers supply services to a variety of customers. In this arrangement, storage backup and retrieval services are given for free, as a subscription, or on a per-user basis. Example: Google App Engine etc.

Advantages of Public Cloud Model:

- Minimal Investment: Because it is a pay-per-use service, there is no substantial upfront fee, making it excellent for enterprises that require immediate access to resources.
- No setup cost: The entire infrastructure is fully subsidized by the cloud service providers, thus there is no need to set up any hardware.
- Infrastructure Management is not required: Using the public cloud does not necessitate infrastructure management.
- No maintenance: The maintenance work is done by the service provider (Not users).
- Dynamic Scalability: To fulfil company's needs, on-demand resources are accessible.

Disadvantages of Public Cloud Model:

- Less secure: Public cloud is less secure as resources are public so there is no guarantee of highlevel security.
- Low customization: It is accessed by many public so it can't be customized according to personal requirements.

2.Private Cloud

The private cloud deployment model is the exact opposite of the public cloud deployment model. It's a one-on-one environment for a single user (customer). There is no need to share your hardware with anyone else. The distinction between private and public clouds is in how you handle all of the hardware. It is also called the "internal cloud" & it refers to the ability to access systems and services within a given border or organization. The cloud platform is implemented in a cloud-based secure environment that is protected by powerful firewalls and under the supervision of an organization's IT department. The private cloud gives greater flexibility of control over cloud resources.

Advantages of Private Cloud Model:

- Better Control: You are the sole owner of the property. You gain complete command over service integration, IT operations, policies, and user behaviour.
- Data Security and Privacy: It's suitable for storing corporate information to which only authorized staff have access. By segmenting resources within the same infrastructure, improved access and security can be achieved.
- Supports Legacy Systems: This approach is designed to work with legacy systems that are unable to access the public cloud.

• Customization: Unlike a public cloud deployment, a private cloud allows a company to tailor its solution to meet its specific needs.

Disadvantages of Private Cloud Model:

• Less scalable: Private clouds are scaled within a certain range as there is less number of clients.

• Costly: Private clouds are costlier as they provide personalized facilities.

3. Hybrid Cloud

By bridging the public and private worlds with a layer of proprietary software, hybrid cloud computing gives the best of both worlds. With a hybrid solution, you may host the app in a safe environment while taking advantage of the public cloud's cost savings. Organizations can move data and applications between different clouds using a combination of two or more cloud deployment methods, depending on their needs.

Advantages of Hybrid Cloud Model:

• Flexibility and control: Businesses with more flexibility can design personalized solutions that meet their particular needs.

• Cost: Because public clouds provide scalability, you'll only be responsible for paying for the extra capacity if you require it.

• Security: Because data is properly separated, the chances of data theft by attackers are considerably reduced.

Disadvantages of Hybrid Cloud Model:

• Difficult to manage: Hybrid clouds are difficult to manage as it is a combination of both public and private cloud. So, it is complex.

• Slow data transmission: Data transmission in the hybrid cloud takes place through the public cloud so latency occurs.

10.b Discuss any five ethical challenges in AI. 10 Marks

1. Bias and Discrimination: AI systems can perpetuate and even amplify biases and discrimination if they are not properly designed and tested. For example, an AI system that is used to make decisions about hiring or lending may discriminate against certain groups of people if it is trained on data that contains such biases.

2. Privacy and Security: AI systems can collect and process large amounts of personal data, which can raise concerns about privacy and security. For example, an AI system that is used to monitor people's behaviour or predict their behaviour may collect sensitive information about them, which could be used to discriminate against them or to cause harm.

3. Lack of Explain ability and Transparency: Many AI systems are based on complex algorithms that are difficult to understand or explain. This can make it difficult for people to understand how the AI system is making its decisions and to hold the system accountable for its actions.

4. Job Loss: AI systems can automate many tasks that are currently performed by humans, which can lead to job loss and other economic dislocation. It's important to consider the social and economic impacts of AI and to ensure that the benefits of AI are shared fairly.

5. Autonomy and Control: AI systems can be programmed to make decisions and take actions autonomously, which can raise concerns about who is in control of the system and who is responsible for its actions.

6. Ethical dilemmas: AI systems may be faced with ethical dilemmas, such as the trade-off between human lives and property damage in self-driving cars, and it may be difficult for the system to make the right decision.

7. Societal impact: The development and deployment of AI can have a significant impact on society, and it's important to consider the broader ethical, social, and political implications of AI and to ensure that the technology.