**Chapter - 1**

1. **Explain Machine learning is a subset of artificial intelligence and a superset of deep learning, with an example.**

**Ans)** Machine learning (ML) is a subset of artificial intelligence (AI), focused on building systems that learn from data to make predictions or decisions without being explicitly programmed for the task. It encompasses a wide range of techniques, including but not limited to deep learning. Deep learning, a subset of machine learning, involves neural networks with many layers, enabling the handling of complex tasks like image and speech recognition. For example, a basic machine learning model might predict housing prices based on features like size and location. In contrast, a deep learning model could analyze images of houses to identify features (e.g., garden, pool) directly from the visuals, providing a more nuanced assessment. Thus, while all deep learning is machine learning, not all machine learning employs deep learning techniques.

1. **Define Machine learning with examples. (Include one more definition in your answer by doing a general internet search and/or from other resources)**

**Ans)** Machine learning is a subset of AI, which is a broad branch of computer science for preparing machines (computers) that can do human tasks.

Machine learning is a field of artificial intelligence that enables computers to learn from and make decisions based on data. Unlike traditional programming, where tasks are explicitly coded, machine learning algorithms adjust their processing based on patterns and insights derived from data. For example, a machine learning model could recommend movies to users based on their viewing history, similar to Netflix's recommendation system. Another example is spam detection in email services, where the system learns to filter out spam by recognizing patterns in messages previously flagged as spam. These examples illustrate how machine learning can adapt to varied and complex tasks by learning from examples and experiences.

1. **Describe two business use cases of ML. (other than above in slides)**

**Ans)**

**a) Customer Segmentation :** Businesses use machine learning to analyze customer data and identify distinct groups based on purchasing behavior, preferences, and demographics. This enables personalized marketing strategies, improving engagement and conversion rates. For instance, an e-commerce company can use ML to segment customers into groups interested in specific product categories, and tailor email marketing campaigns to each segment, increasing relevancy and sales.

**b) Predictive Maintenance :** In manufacturing, machine learning algorithms analyze data from machinery sensors to predict equipment failures before they happen. This proactive approach allows companies to schedule maintenance efficiently, reducing downtime and operational costs. For example, a manufacturing plant could use ML to monitor the condition of its equipment, predict which machines are likely to fail and when, and perform maintenance only when necessary, optimizing the use of resources and minimizing interruptions.

1. **List types of ML, with one example of each.**

**Ans)**

1. **Supervised Learning :** This involves learning a function that maps an input to an output based on example input-output pairs. An example is a spam detection system where emails are labeled as "spam" or "not spam," and the algorithm learns to classify unseen emails.
2. **Unsupervised Learning :** Here, the algorithm learns patterns from untagged data. A common example is customer segmentation in marketing, where customers are grouped into clusters based on similarities in their purchasing behavior without predefined labels.
3. **Reinforcement Learning :** This type involves learning to make decisions by taking actions in an environment to maximize some notion of cumulative reward. An example is a chess game AI, which learns optimal moves by playing many games against itself or opponents and improving over time based on wins and losses.
4. **Explain Unsupervised Learning in detail with any use case.**

**Ans)** Unsupervised learning involves training machine learning models on data without explicit instructions on what to predict. The model identifies patterns, structures, or insights from the input data without any labeled responses to guide the process. A key use case is market basket analysis, commonly used in retail to uncover associations between different products. By analyzing transaction data, an unsupervised learning model can identify which products are frequently bought together, enabling retailers to optimize product placement and cross-promotional strategies. This approach helps in increasing sales by strategically recommending products to customers based on their shopping patterns.

1. **Explain Supervised Learning in detail with any use case.**

**Ans)** Supervised learning is a type of machine learning where an algorithm is trained on labeled data, meaning the algorithm is provided with input-output pairs. The goal is for the model to learn to predict the output from new, unseen inputs based on the training data. A classic use case is email spam detection, where emails are labeled as "spam" or "not spam," and the model is trained on these labels to classify new emails. The algorithm adjusts its parameters to minimize errors in predictions, improving its accuracy over time. By applying this technique, email services can effectively filter out unwanted emails, enhancing user experience by reducing the amount of spam that reaches their inbox.

1. **List phases of standard Machine Learning pipeline process with diagram.**

**Ans)**

1. **Phase 1 - Problem Formulation :** Define what you are trying to solve; this guides the entire process.
2. **Phase 2 - Collect and Label Data :** Assemble the dataset you’ll use for training the model. If the model is supervised, this step includes labeling the data.
3. **Phase 3 - Evaluate Data :** Assess the quality and usefulness of the collected data, potentially returning to Phase 2 to refine the data collection process if needed.
4. **Phase 4 - Feature Engineering :** Select and possibly transform the most informative attributes (features) that contribute to the prediction or classification task.
5. **Phase 5 - Select and Train Model :** Choose an appropriate machine learning algorithm and train it with the data. This involves adjusting the model’s parameters to best fit the data.
6. **Phase 6 - Deploy Model :** Once the model is adequately trained and tested, deploy it into a production environment where it can provide predictions or classifications on new data.
7. **Phase 7 - Evaluate Model :** Continuously measure the model’s performance to ensure it meets the predefined success criteria. This step can lead to revisiting previous stages to refine the model.
8. **Phase 8 - Tune Model :** Based on performance evaluations, further adjust and optimize the model's hyperparameters to improve its accuracy or ability to generalize.
9. **Describe any two ML pipeline processes in detail, using any use case.**

**Ans)**

1. **Feature Engineering :** In a credit scoring use case, feature engineering involves selecting and transforming borrower data like income, employment history, and credit history into formats that a machine learning model can use effectively. Engineers might create new features such as debt-to-income ratio or number of late payments in the past year, providing nuanced variables that more accurately predict creditworthiness.
2. **Model Training :** For a product recommendation system, model training would involve using historical user purchase data and item attributes to train a model. The model learns patterns of user behavior and item affinity. For instance, using a collaborative filtering algorithm, the system identifies products to recommend based on a user's previous interactions and the interactions of others with similar tastes.
3. **Describe ML challenges.**

**Ans)**

1. **Data Quality :** Poor quality data with noise, errors, or biases can significantly degrade the performance of machine learning models.
2. **Data Quantity :** Machine learning algorithms often require large volumes of data to perform well, and acquiring such datasets can be difficult and expensive.
3. **Overfitting :** Models can become too complex, capturing noise as if it were a significant pattern, which leads to poor performance on new, unseen data.
4. **Underfitting :** Conversely, overly simple models may not capture the underlying trends in the data, leading to inadequate performance.
5. **Computational Costs :** Training more sophisticated models requires substantial computational resources and time, which can be costly.
6. **Interpretability :** Complex models, like deep neural networks, can act as "black boxes," making it hard to understand how they make decisions, which is a problem for applications that require transparency.
7. **Generalization :** Ensuring that a model performs well across different datasets and real-world scenarios is a constant challenge.
8. **Ethical Concerns :** Mitigating bias and ensuring that models do not perpetuate or exacerbate unfair biases present in the training data is an ongoing concern.
9. **Which types of roles (jobs) in the area of ML, you may perform in future?**

**Ans)**

1. **Data Scientist :** A role that combines statistical analysis, machine learning, and data visualization to turn data into actionable insights, driving strategic decisions across various sectors.
2. **Machine Learning Engineer :** Focused on designing and creating ML systems and pipelines, ensuring they are scalable and integrated with other software applications.
3. **Applied Science Researcher :** Working typically in industrial labs, applying ML theories to solve practical, real-world problems and often moving scientific findings closer to tangible products.
4. **Machine Learning Developer :** Specializing in coding and deploying machine learning models, often with a stronger emphasis on software development practices and less on the research aspect.

**Chapter - 2**

1. **Create a DataFrame named temperatures from a dictionary of three temperature readings each for 'Maxine’, 'James' and 'Amanda’.**

**Ans)**

import pandas as pd

temperature\_data = {

'Maxine': [25, 28, 22],

'James': [22, 24, 20],

'Amanda': [26, 30, 24]

}

temperatures = pd.DataFrame(temperature\_data)

print("Temperatures DataFrame:")

print(temperatures)

1. **Recreate the DataFrame temperatures in Part (a) with custom indices using the index keyword argument and a list containing 'Morning', 'Afternoon' and 'Evening’.**

**Ans)**

temperatures\_custom\_indices = pd.DataFrame(temperature\_data, index=['Morning', 'Afternoon', 'Evening'])

print("\nTemperatures DataFrame with Custom Indices:")

print(temperatures\_custom\_indices)

1. **Select from temperatures the column of temperature readings for 'Maxine’.**

**Ans)**

maxine\_temperatures = temperatures['Maxine']

print("\nMaxine's Temperatures:")

print(maxine\_temperatures)

1. **Select from temperatures the row of 'Morning’ temperature readings.**

**Ans)**

morning\_temperatures = temperatures\_custom\_indices.loc['Morning']

print("\nMorning Temperatures:")

print(morning\_temperatures)

1. **Select from temperatures the rows for 'Morning' and 'Evening' temperature readings.**

**Ans)**

morning\_evening\_temperatures = temperatures\_custom\_indices.loc[['Morning', 'Evening']]

print("\nMorning and Evening Temperatures:")

print(morning\_evening\_temperatures)

1. **Select from temperatures the columns of temperature readings for 'Amanda' and 'Maxine’.**

**Ans)**

amanda\_maxine\_temperatures = temperatures\_custom\_indices[['Amanda', 'Maxine']]

print("\nAmanda and Maxine's Temperatures:")

print(amanda\_maxine\_temperatures)

1. **Select from temperatures the elements for 'Amanda' and 'Maxine' in the 'Morning' and 'Afternoon’.**

**Ans)**

amanda\_maxine\_morning\_afternoon\_temperatures = temperatures\_custom\_indices.loc[['Morning', 'Afternoon'], ['Amanda', 'Maxine']]

print("\nAmanda and Maxine's Morning and Afternoon Temperatures:")

print(amanda\_maxine\_morning\_afternoon\_temperatures)

1. **Use the describe method to produce temperatures’ descriptive statistics.**

**Ans)**

temperatures\_descriptive\_stats = temperatures\_custom\_indices.describe()

print("\nTemperatures Descriptive Statistics:")

print(temperatures\_descriptive\_stats)

1. **Transpose of temperatures.**

**Ans)**

temperatures\_transposed = temperatures\_custom\_indices.transpose()

print("\nTransposed Temperatures:")

print(temperatures\_transposed)

1. **Sort temperatures so that its column names are in alphabetical order.**

**Ans)**

temperatures\_sorted = temperatures\_custom\_indices.sort\_index(axis=1)

print("\nSorted Temperatures:")

print(temperatures\_sorted)

1. **Class Average: Writing Grades to a Plain Text File. In an Ipython session, write code that enables you to store any number of grades into a grades.txt plain text file.**

**Ans)**

import csv

def write\_grades\_to\_file(grades):

with open('grades.txt', 'w') as file:

for grade in grades:

file.write(str(grade) + '\n')

# Example grades

grades = [85, 90, 75, 95, 80]

write\_grades\_to\_file(grades)

1. **Class Average: Reading Grades from a Plain Text File. In an Ipython session, write code that reads the grades from the grades.txt file you created in the previous exercise. Display the individual grades and their total, count and average.**

**Ans)**

def read\_grades\_from\_file(file\_name):

with open(file\_name, 'r') as file:

grades = [int(line.strip()) for line in file]

return grades

grades = read\_grades\_from\_file('grades.txt')

print("Individual Grades:", grades)

print("Total:", sum(grades))

print("Count:", len(grades))

print("Average:", sum(grades) / len(grades))

1. **Class Average: Writing Student Records to a CSV File. An instructor teaches a class in which each student takes three exams. The instructor would like to store this information in a file named grades.csv for later use. Write code that enables an instructor to enter each student’s first name and last name as strings and the student’s three exam grades as integers. Use the csv module to write each record into the grades.csv file. Each record should be a single line of text in the following CSV format:**

***firstname, lastname, exam1grade, exam2grade, exam3grade***

**Ans)**

def write\_student\_records\_to\_csv(students):

with open('grades.csv', 'w', newline='') as file:

writer = csv.writer(file)

writer.writerow(['Firstname', 'Lastname', 'Exam1', 'Exam2', 'Exam3'])

for student in students:

writer.writerow(student)

# Example student records

students = [['John', 'Doe', 85, 90, 75],

['Jane', 'Smith', 95, 80, 85]]

write\_student\_records\_to\_csv(students)

1. **Class Average: Reading Student Records from a CSV File. Use the csv module to read the grades.csv file from the previous exercise. Display the data in tabular format.**

**Ans)**

def read\_student\_records\_from\_csv(file\_name):

with open(file\_name, 'r', newline='') as file:

reader = csv.reader(file)

for row in reader:

print(row)

# Read student records from file

read\_student\_records\_from\_csv('grades.csv')

1. **Class Average: Creating a Grade Report from a CSV File. Modify your solution to the preceding exercise to create a grade report that displays each student’s average to the right of that student’s row and the class average for each exam below that exam’s column.**

**Ans)**

def create\_grade\_report(file\_name):

with open(file\_name, 'r', newline='') as file:

reader = csv.reader(file)

data = list(reader)

for row in data:

row.extend([sum(map(int, row[2:])), sum(map(int, row[2:])) / 3])

class\_avg = [sum(map(int, [row[i] for row in data[1:]])) / len(data[1:]) for i in range(2, 5)]

for avg in class\_avg:

data.append(['Class Avg:', '', '', avg, ''])

for row in data:

print(row)

# Create grade report

create\_grade\_report('grades.csv')

1. **In IPython session, load the grades.csv file and then display the dataframe.**

**Ans)**

import pandas as pd

df = pd.read\_csv('grades.csv')

print(df)

1. **Write a python program for the temperature prediction for January month at New Your City, (In Python script number 10.16), Assuming that this linear trend continues, based on the slope and intercept values calculated in this Python script section's interactive session, in what year might the average January temperature in New York City reach 40 degrees Fahrenheit.**

**Ans)**

# Slope and intercept values from the previous linear regression analysis

slope\_temperature = 1.8333333333333335

intercept\_temperature = 18.666666666666664

# Target temperature in Fahrenheit

target\_temperature\_fahrenheit = 40

# Convert Fahrenheit to Celsius using the formula: °C = (°F - 32) \* 5/9

target\_temperature\_celsius = (target\_temperature\_fahrenheit - 32) \* 5/9

# Predict the temperature for January using the linear regression equation

predicted\_temperature\_celsius = slope\_temperature \* 13 + intercept\_temperature

# Calculate the number of years until the predicted temperature reaches the target temperature

years\_until\_target\_temperature = abs((target\_temperature\_celsius - predicted\_temperature\_celsius) / slope\_temperature)

# Calculate the predicted year

current\_year = 2024 # Assume the current year

predicted\_year = current\_year + int(years\_until\_target\_temperature)

# Output the predicted year

print(f"The average January temperature in New York City is predicted to reach {target\_temperature\_fahrenheit}°F in the year {predicted\_year}.")

**Chapter – 3**

1. **Describe ML Model case components – 1) question (business problem), 2) why, and 3) dataset. With at least one example.**

**Ans)** The question in an ML model scenario specifies the business issue that needs to be resolved, such forecasting loan default. The "why"—such as reducing financial risk—explains the importance of the issue. Relevant data points, including credit histories of borrowers, are included in the dataset. For instance, using a dataset of previous loan applications and repayment histories, a bank may use an ML model to estimate the likelihood of loan default, which is critical for risk management and profitability.

1. **Which are ML pipeline processes phase-2 and phase-3? Explain briefly.**

**Ans)** The "Evaluate data" phase of the ML pipeline is when anomalies are found, insights are obtained, and the quality and use of the data are evaluated. "Feature engineering" (Phase 3) entails using domain expertise and methods like one-hot encoding, binning, and normalization to create new features from the available data in order to enhance model performance.

1. **How to collect data from the data source and what type of data source is available, explain with an example.**

**Ans)** Data can be gathered from commercial sources provided by businesses like Reuters or Dun & Bradstreet, or from private sources that are kept internally by an organization, such as bank transaction records. One way to gather data would be to subscribe to a Reuters dataset via the AWS Data Exchange. This dataset would include international financial news that financial institutions might use to guide their investment decisions.

1. **Describe the ETL process with any ETL tool/ service of your choice.**

**Ans)** Extract, Transform, Load, or ETL, is a process used in data processing. 'Extract' is the first step in the process, when data is pulled from multiple sources, including databases and APIs, using an ETL tool like Apache NiFi. After this data has been cleaned, enriched, and transformed into a format appropriate for analysis, 'Transform' may involve filtering, sorting, and aggregating the data. The final step, "load," is moving the changed data into a database or data warehouse, such as Amazon Redshift. With its user-friendly interface and ability to automate data flows, Apache NiFi shines in this process, guaranteeing that the data is dependable and prepared for business intelligence tasks.

1. **Describe how you collect your data with the data frame of Python pandas. Explain with an example.**

**Ans)** Using built-in functions in pandas for Python, data is gathered into a DataFrame by reading from a data source. For instance, `pandas.read\_csv('file.csv')} would use a CSV file to construct a DataFrame. This DataFrame functions as a table-like structure that facilitates effective data analysis and manipulation. Additionally, you can use `pandas.read\_sql(query, connection)} to gather data from databases and `pandas.read\_excel('file.xlsx')} to gather data from Excel files. These sources can be converted straight into pandas DataFrames and used right away in your data analysis processes.

1. **Describe descriptive statistics with categories.**

**Ans)** The characteristics of a dataset are summed up and described using descriptive statistics. measurements of variability and measurements of central tendency are the two categories into which they fall. The center point of a data distribution is represented by the mean, median, and mode, which are examples of central tendency. Variability measurements characterize the distribution and dispersion of data points. Examples include standard deviation, variance, range, and interquartile range. Before conducting further in-depth research, these statistics are essential for comprehending the core characteristics of the data.

1. **How pandas describe() function help you in phase-2 and phase-3 of the ML pipeline process.**

**Ans)** Pandas' `describe()` function summarizes the form, dispersion, and central tendency of a dataset's distribution, excluding NaN values, for phase-2, 'Evaluate data'. For a fast statistical analysis of numerical columns, it is helpful. {describe()} may be used to help discover features with substantial deviations in phase 3, or "feature engineering," which may help with normalization or standardization procedures. It can also be used to indicate the need for new features to be created or current ones to be altered in order to better capture the properties of the data.

1. **What is a histogram? And how it helps us in the data evaluation process of the ML pipeline.**

**Ans)** A histogram is a graphical depiction of a numerical dataset's frequency distribution using bars. The amount of data points that fall into each bin is shown by the height of each bar, which classifies the data into bins. Histograms aid in the comprehension of variable distribution, the detection of outliers, the identification of modality, and the guidance of essential data modifications, such as normalization or standardization, to enhance model performance during the data review phase of an ML pipeline.

1. **Explain the use of correlation matrix and heat map.**

**Ans)** The degree of linear relationship between variables in a dataset is measured by a correlation matrix, which has values ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation). This matrix is shown using a heatmap, which highlights strong or weak associations more easily by using colors to represent the correlation values. This helps with feature selection in machine learning by drawing attention to possible redundancy (high correlation) across features and by locating possible predictors that have significant correlations to the target variable.

**Chapter – 4**

1. **What is the ML pipeline process phase-4? Explain briefly.**

**Ans)** Phase 4 of the ML pipeline process is "Feature Engineering," which is a critical step involving the selection, manipulation, and transformation of raw data into features that can effectively represent the underlying problem to the predictive models. This process improves model accuracy by creating feature sets from the raw data that highlight important patterns for learning. For instance, in a fraud detection system, feature engineering might involve creating new features such as the frequency of transactions in a certain time period, the average transaction amount, or the ratio of online to in-person transactions. Effective feature engineering can often make or break a model's performance on a given task.

1. **Describe Encoding ordinal data with an example.**

**Ans)** Encoding ordinal data involves converting categorical data values that have a natural order into numerical form while maintaining the order. For example, consider T-shirt sizes: Small, Medium, Large, and Extra Large. These sizes have a natural ordering from smallest to largest. To encode these, we might assign Small to 0, Medium to 1, Large to 2, and Extra Large to 3. This numerical encoding maintains the relative ordering of the sizes, which can then be used in machine learning models to predict outcomes that depend on this ordinal relationship, such as customer preferences or stock levels.

1. **What is the “Cleaning data” concept? Explain with an example.**

**Ans)** Cleaning data is the process of fixing or removing incorrect, corrupted, duplicate, or incomplete information within a dataset. For instance, if a dataset of survey results contains multiple spellings for a respondent's gender such as "Female," "female," "F," and "fem.," data cleaning would involve standardizing these entries into a single, consistent format. This might mean transforming all variations to "Female." This process ensures that the data is accurate and uniform, which is crucial for effective analysis and machine learning model training, as inconsistencies and errors can lead to incorrect conclusions and poor model performance.

1. **Explain the “drop or impute” missing value with the scenario of each.**

**Ans**) "Drop or impute" are two strategies for handling missing values in a dataset:

**Dropping :** If a dataset has a high number of missing values in a particular feature, or if the missing data is not random, you might choose to drop those records or the entire feature to maintain the integrity of the model. For example, if most income data in a loan eligibility dataset is missing, that column might be dropped to avoid bias.

**Imputing :** Alternatively, missing values can be filled in, or 'imputed', with statistical measures like mean, median, or mode, or through more complex algorithms that predict missing values based on other data. For instance, missing values in a temperature dataset might be imputed using the average temperature from other days with similar conditions. This maintains the dataset's size and can improve model performance if done carefully.

1. **What are outliners and how you can deal with them?**

**Ans)** Outliers are data points that differ significantly from other observations in a dataset, potentially due to variability in the measurement or experimental errors. They can skew and mislead the training process of machine learning models, resulting in a loss of accuracy. To deal with outliers, one can:

**a) Remove them :** If they are a result of errors or noise, they can be deleted.

**b) Cap them :** Use a threshold to cap values at a certain percentile.

**c) Transform them :** Apply a mathematical transformation to reduce the skewness.

**d) Impute them :** Replace outliers with statistically derived values.

**e) Keep them :** If they are legitimate values, they might be essential to keep for the model to learn from exceptions.

The strategy chosen depends on the nature of the data and the analytical goals.

1. **Write notes on feature selection methods.**

**Ans)** Feature selection methods are used to identify and select the most relevant features to use in model construction, which can lead to improved model accuracy, reduced overfitting, and decreased computational costs. Common methods include:

1. **Filter Methods :** These methods apply a statistical measure to assign a scoring to each feature; features are selected or removed based on their scores. For example, the chi-squared test can be used for categorical data.
2. **Wrapper Methods :** These use a subset of features and train a model using them, evaluating the model performance to decide whether to keep or discard a feature, like in recursive feature elimination.
3. **Embedded Methods :** These incorporate feature selection as part of the model training process and include techniques like LASSO, which adds a penalty to the loss function to reduce the coefficients of less important features to zero.