Assessment Questions:

1. Explain how you would handle missing data in a given dataset and provide a code snippet demonstrating this.

To handle missing data in a dataset, there are several approaches you can take depending on the nature and extent of the missing data. Here are a few common techniques:

- Deleting rows: If the missing data is relatively small in quantity and randomly distributed, you can choose to delete the rows with missing values. However, this approach should be used with caution as it may lead to a loss of valuable information.

- Imputation: Imputation involves replacing missing values with estimated values based on the available data. There are various methods for imputation, such as mean imputation (replacing missing values with the mean of the column), median imputation (replacing missing values with the median of the column), or regression imputation (predicting missing values using regression models).

- Indicator variables: In some cases, it might be meaningful to create an additional binary column that indicates whether a value is missing or not. This approach allows the missingness to be explicitly incorporated into the analysis.

Here's an example code snippet using Python and pandas library to demonstrate mean imputation:

```
import pandas as pd

# Load the dataset
df = pd.read_csv('your_dataset.csv')

# Check for missing values
missing_values = df.isnull().sum()
```

```
# Impute missing values with mean
df_filled = df.fillna(df.mean())

# Verify if missing values are handled
missing_values_after_imputation = df_filled.isnull().sum()

# Display the filled dataset
print(df_filled.head())
```

This code snippet loads a dataset from a CSV file, checks for missing values, and replaces the missing values with the mean of each respective column using the fillna() function from pandas.

2. Prepare a high-level lesson plan for an introductory session on deep learning.

High-level lesson plan for an introductory session on deep learning:

- Introduction to Neural Networks: Explain the basic concept of artificial neural networks, including neurons, activation functions, and weights.
- Deep Learning vs. Traditional Machine Learning: Discuss the key differences between deep learning and traditional machine learning algorithms.
- Deep Learning Architectures: Introduce popular deep learning architectures like feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs).
- Training Deep Learning Models: Explain the process of training deep learning models using backpropagation and gradient descent.
- Deep Learning Applications: Discuss real-world applications of deep learning, such as image recognition, natural language processing, and autonomous vehicles.
- Hands-on Exercise: Provide a coding exercise using a deep learning framework like TensorFlow or PyTorch to build a simple neural network for a classification task.
- Q&A and Wrap-up: Allow time for questions and summarize the key takeaways from the session.

3. How would you troubleshoot a machine learning model whose performance isn't as expected? Discuss your approach briefly.

Troubleshooting a machine learning model with unexpected performance:

- Data Inspection: Start by examining the data used for training and evaluation. Look for any data quality issues, missing values, or outliers that might impact the model's performance.
- Feature Selection and Engineering: Evaluate the features used in the model. Consider whether they are relevant, informative, and properly encoded for the problem at hand.
- Model Evaluation: Assess the model's performance metrics, such as accuracy, precision, recall, or F1 score. Identify which metrics are not meeting expectations.
- Error Analysis: Examine misclassified or poorly predicted samples. Look for patterns or common characteristics that might indicate areas for improvement.
- Hyperparameter Tuning: Adjust the hyperparameters of the model to find the optimal configuration. This can involve grid search, random search, or more advanced optimization techniques.
- Model Complexity: Evaluate whether the model is too simple or too complex for the problem. Adjust the model architecture or consider regularization techniques to find the right balance.
- Dataset Size: Check if the dataset is large enough to capture the complexity of the problem. Consider gathering more data or applying data augmentation techniques.
- Cross-Validation: Validate the model's performance using cross-validation to ensure that the results are consistent and not dependent on the specific training-validation split.
- Model Selection: If multiple models are available, compare their performance and choose the one that best suits the problem.
- Continuous Learning: Keep track of the model's performance over time and monitor for any concept drift. Retrain or update the model periodically as new data becomes available.

4. Explain in simple terms what Natural Language Processing (NLP) is and its real-world applications.

Natural Language Processing (NLP) is a branch of artificial intelligence that focuses on enabling computers to understand, interpret, and generate human language. It involves developing algorithms and models that can process and analyse textual data in a way that is like how humans understand language.

Real-world applications of NLP include:

- Sentiment Analysis: Determining the sentiment or opinion expressed in a piece of text, often used in social media monitoring, customer feedback analysis, and brand reputation management.
- Machine Translation: Translating text from one language to another, such as Google Translate, allowing communication across language barriers.
- Chatbots and Virtual Assistants: Building conversational agents that can understand and respond to natural language queries, used in customer support, virtual assistants, and automated messaging systems.
- Named Entity Recognition (NER): Identifying and classifying named entities (such as names, locations, organizations) within text, helpful in information extraction and text summarization.
- Text Classification: Categorizing text into predefined classes or topics, used in spam detection, sentiment analysis, news categorization, and document classification.
- Language Generation: Generating human-like text, including chat responses, product descriptions, and personalized recommendations.

5. Write a SQL query to retrieve specific information from a relational database.

Sort by Text Title and then by alphanumeric title and then by numeric title within each group,sort by title again

```
select Title,
if(title not regexp '[0-9]', 'Text Title',if(title regexp '[A-z]', 'Aplanumeric Title','Numeric Title'))as 'TitleType'
from movies.film
order by case
when TitleType = 'Text Title' then 1
when TitleType = 'Aplanumeric Title' then 2
else 3
end asc, title asc;
```