1. What is Machine Learning and how does it differ from traditional programming?

Machine Learning is a subset of Artificial Intelligence that enables systems to learn from data and improve their performance on specific tasks without being explicitly programmed. Instead of writing explicit rules, it identifies patterns in data and uses them to make decisions or predictions. In traditional programming, developers write explicit rules and instructions for the system to follow, whereas in Machine Learning, the system learns rules and patterns from the data. Developers provide algorithms and datasets for the system to learn from.

1. Discuss the relationship between Artificial Intelligence, Machine Learning and Deep Learning.

Artificial Intelligence (AI) is the broadest field that involves mimicking human intelligence to perform tasks like reasoning, learning, and problem-solving.

Machine Learning (ML), a subset of AI, allows systems to learn from data and improve performance on specific tasks without explicit programming.

Deep Learning (DL), a subset of ML, uses neural networks with multiple layers to model complex patterns in large datasets, excelling in tasks like image recognition and natural language processing.

1. List some common applications of Machine Learning in various industries.

Machine learning is widely applied in healthcare for tasks like disease diagnosis, personalized medicine, and drug discovery, such as detecting cancer from medical images.

In finance, it helps with fraud detection, stock price prediction, and risk assessment, like identifying fraudulent credit card transactions.

In retail, it powers recommendation systems, customer behavior analysis, and inventory management, exemplified by Amazon's personalized product recommendations.

1. What are the main differences between supervised and unsupervised learning? Provide examples of each.

Supervised learning learns from labeled data, where input-output pairs are known, and includes tasks like classification and regression.

Unsupervised learning works with unlabeled data, finding patterns or structures, and includes methods like clustering and dimensionality reduction.

1. Explain the steps involved in a typical machine learning pipeline.
   1. Problem definition: Define the problem, objective and the desired outcomes
   2. Data Collection: Gather relevant data from various sources
   3. Data Preprocessing: Clean, transform and format the data
   4. Exploratory Data Analysis: Analyse the data to understand its structure, identify patterns
   5. Feature Engineering: Select, create or transform features that will be used by the model to make predictions
   6. Model Selection: Choose an appropriate machine learning algorithm based on the type of problem
   7. Model Training: Train the model based on the prepared data allowing it to learn patterns from the input data
   8. Model Evaluation: Evaluate the model performance using evaluation metrics
   9. Hyperparameter Tuning: Find and tune the model’s hyperparameter to improve its performance and efficiency
   10. Deployment: Once the model is trained and evaluated, deploy it in a real-world application.
   11. Monitoring and Maintenance: Continuously monitor the model’s performance in production, making updates or retraining when necessary to adapt to new data or changing conditions
2. What is the difference between classification and regression in machine learning? Give examples of each.

Classification is a machine learning task where the goal is to predict discrete categories or labels, such as predicting whether an email is "spam" or "not spam."

Regression involves predicting continuous numerical values, such as estimating house prices based on various features.

1. Discuss the difference between training sets and test sets in machine learning. Why is it important to separate these datasets?

Training Set: The training set is the portion of the dataset used to train the machine learning model. The model learns patterns, relationships, and features from this data to make predictions.

Test Set: The test set is a separate portion of the dataset that is not used during training. It is used to evaluate the performance of the trained model and ensure it generalizes well to unseen data.

Separating the datasets helps to prevent **overfitting**, where the model memorizes the training data instead of learning general patterns. The test set provides an unbiased evaluation of the model’s performance on new, unseen data.

1. What are some common challenges or limitations of machine learning algorithms? How can these challenges be mitigated?

**Overfitting**: This occurs when the model learns the noise or irrelevant details in the training data, making it perform poorly on new data.

**Mitigation**: Use techniques like regularization (e.g., L1/L2 regularization), cross-validation, and pruning (in decision trees) to reduce overfitting. Simplifying the model and using more data can also help.

**Underfitting**: This happens when the model is too simple to capture the underlying patterns in the data, resulting in poor performance on both training and test data.

**Mitigation**: Use more complex models, add more features, or improve data preprocessing to allow the model to learn more effectively.

**Insufficient** **Data**: A lack of sufficient data can limit the model's ability to learn properly, leading to poor generalization.

**Mitigation**: Gather more data, apply data augmentation techniques, or use transfer learning (leveraging pre-trained models) to overcome the data limitation.

**Bias** **and** **Variance**: High bias leads to underfitting, while high variance leads to overfitting. Balancing these is essential.

**Mitigation**: Use the bias-variance tradeoff to find a balance, adjust the complexity of the model, and use cross-validation to check performance on different datasets.

**Data** **Quality**: Poor-quality or noisy data can lead to inaccurate predictions and poor model performance.

**Mitigation**: Clean and preprocess the data properly by handling missing values, removing outliers, and normalizing features to improve data quality.

**Interpretability**: Complex models like deep learning can act as "black boxes," making it difficult to understand why a model makes a particular decision.

**Mitigation**: Use simpler, more interpretable models (e.g., decision trees), or apply explainability techniques such as SHAP or LIME to interpret the model’s decisions.