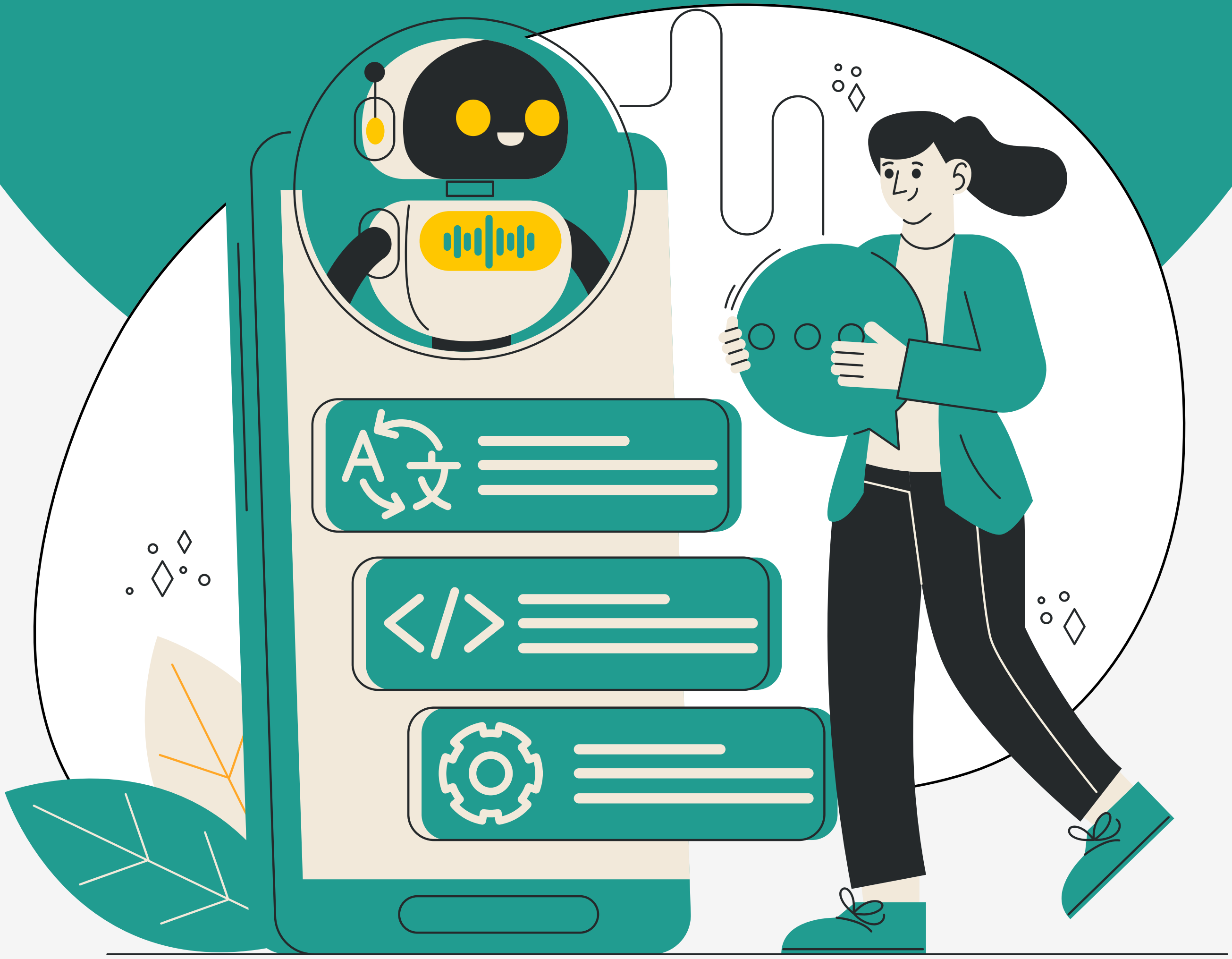
Introduction to Machine Learning

Interview Questions

(Practice Project)



PW SKILLS

1. **What is machine learning, and how does it differ from traditional programming?**

Ans: Machine learning (ML) is a subset of artificial intelligence that enables computers to learn from data and improve their performance over time without being explicitly programmed for each task. In contrast, traditional programming relies on predefined rules and logic crafted by developers. While traditional programming requires programmers to anticipate every possible scenario and write specific instructions, machine learning models learn from data, allowing them to adapt to new situations and make predictions based on patterns identified in the data. This flexibility makes ML particularly suitable for complex tasks involving large datasets, such as image recognition and natural language processing.

2. **Can you explain the difference between supervised and unsupervised learning with examples?**

Ans: Supervised Learning involves training a model on a labeled dataset, where the input data is paired with the correct output. For example, in a spam detection system, emails are labeled as "spam" or "not spam," allowing the model to learn to classify new emails based on this training.

Unsupervised Learning, on the other hand, deals with unlabeled data. The model tries to find hidden patterns or intrinsic structures in the input data. A common example is customer segmentation in marketing, where the algorithm groups customers based on purchasing behavior without predefined labels.

3. **What are the different types of machine learning, and in which scenarios would you use each?**

Ans: Types of machine learning algorithms are:

1. **Supervised Learning:** Used when you have labeled data and want to predict outcomes, such as classification tasks (e.g., identifying if an email is spam) or regression tasks (e.g., predicting house prices).
2. **Unsupervised Learning:** Applied when you have unlabeled data and want to explore patterns, such as clustering customers into segments based on behavior.
3. **Reinforcement Learning:** Involves training models to make sequences of decisions by rewarding desired behaviors, often used in robotics and game playing (e.g., training an AI to play chess).
4. **Semi-supervised Learning:** Combines labeled and unlabeled data, useful when labeling data is expensive or time-consuming, such as in image classification tasks where only a few images are labeled.

4. **Describe the process of splitting a dataset into training, validation, and test sets. Why is this important?**

Splitting a dataset involves dividing it into three distinct subsets:

Ans: Process of splitting a dataset is given below:

1. **Training Set:** Used to train the model, allowing it to learn the patterns in the data.
2. **Validation Set:** Used to tune the model's hyperparameters and make decisions about the model's architecture.
3. **Test Set:** Used to evaluate the model's performance on unseen data to estimate how it will perform in real-world scenarios.

This process is crucial because it helps prevent overfitting, ensuring that the model generalizes well to new data rather than just memorizing the training data.

5. What are common ratios for splitting data into training, validation, and test sets?

Ans: Co**mmon ratios for splitting data are:**

- 70% Training, 15% Validation, 15% Test
- 80% Training, 10% Validation, 10% Test

These ratios can vary based on the size of the dataset and the specific requirements of the project.

6. Explain K-Fold cross-validation and why it is used.

Ans: K-Fold cross-validation is a technique used to assess the performance of a model. The dataset is divided into K subsets (or folds). The model is trained on K-1 folds and tested on the remaining fold, repeating this process K times, with each fold serving as the test set once. This method provides a more reliable estimate of model performance by reducing the variance associated with a single train-test split, ensuring that every data point is used for both training and testing.

7. What is overfitting, and how can you prevent it in a machine learning model?

Ans: Overfitting occurs when a model learns the training data too well, capturing noise and outliers instead of the underlying pattern. This results in poor performance on unseen data. To prevent overfitting, you can:

- Use simpler models with fewer parameters.
- Apply regularization techniques (L1 or L2).
- Use cross-validation to ensure the model generalizes well.
- Gather more training data if possible.

8. What are some signs that a model is overfitting?

Ans: Signs of overfitting include:

- High accuracy on the training set but significantly lower accuracy on the validation/test set.
- The model performs well on known examples but poorly on new, unseen data.
- Complex models with high variance that react strongly to small fluctuations in the training data.

9. How would you address underfitting in a model?

Ans: To address underfitting, where a model is too simple to capture the underlying trend, you can:

- Increase model complexity by adding more features or using a more sophisticated algorithm.
- Reduce regularization if it is too strong.
- Ensure that the model is trained for enough epochs to learn from the data.

10. What is the bias-variance tradeoff, and why is it important in machine learning?

Ans: The bias-variance tradeoff refers to the balance between two sources of error in a model:

- **Bias:** Error due to overly simplistic assumptions in the learning algorithm, leading to underfitting.
- **Variance:** Error due to excessive complexity in the model, leading to overfitting.

Understanding this tradeoff is crucial because it helps in selecting the right model complexity and tuning hyperparameters to achieve optimal performance.

11. Can you explain the difference between high bias and high variance in a model?

Ans: Difference between high bias and high variance :

- **High Bias:** Models with high bias are too simplistic, failing to capture the underlying patterns in the data. They tend to underfit, resulting in poor performance on both training and test datasets.
- **High Variance:** Models with high variance are overly complex, capturing noise in the training data. They perform well on training data but poorly on unseen data, leading to overfitting.

12. How can you achieve a balance between bias and variance in a machine learning model?

Ans: To balance bias and variance, one can:

- Choose an appropriate model complexity based on the data.
- Use techniques like cross-validation to evaluate model performance.
- Employ regularization methods to constrain model complexity.
- Experiment with different algorithms to find the best fit for the data.

13. What is regularization, and how does it help in preventing overfitting?

Ans: Regularization is a technique used to discourage overly complex models by adding a penalty term to the loss function. This penalty discourages large coefficients in the model, effectively simplifying it. Regularization helps prevent overfitting by ensuring that the model generalizes better to new data.

14. Why is cross-validation preferred over a simple train-test split?

Ans: Cross-validation is preferred because it provides a more reliable estimate of model performance. A simple train-test split can lead to high variance in performance estimates, depending on how the data is divided. Cross-validation mitigates this by using multiple splits, ensuring that every data point is used for both training and validation, which leads to a more accurate assessment of how the model will perform on unseen data.

15. What are some common causes of underfitting in a model?

Ans: Common causes of underfitting include:

- Using overly simplistic models that cannot capture the complexity of the data.
- Insufficient training time or epochs.
- Lack of relevant features or poor feature selection.
- High levels of regularization that overly constrain the model.