**1. What is the definition of a target function? In the sense of a real-life example, express the target function. How is a target function's fitness assessed?**

A **target function** (also called a **true function**) refers to the underlying function that represents the relationship between input features and the corresponding output (or label). It defines the "true" relationship or mapping in the problem space.

**Real-life example**: Consider predicting the price of a house based on features like size, location, and number of rooms. The target function could be the real-world pricing mechanism that relates these features to the price of the house.

**Fitness Assessment**: The fitness of a target function is usually assessed by comparing the model’s predicted values with the actual values. In machine learning, metrics like **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, or **Accuracy** are used to measure how well the model approximates the target function.

**2. What are predictive models, and how do they work? What are descriptive types, and how do you use them? Examples of both types of models should be provided. Distinguish between these two forms of models.**

* **Predictive Models**: These models forecast future outcomes based on historical data. They learn patterns from past data and make predictions about unknown future data. For example, **linear regression** or **random forests** for predicting house prices.

**Example**: A weather prediction model that uses historical data on temperature, humidity, and wind speed to predict future temperatures.

* **Descriptive Models**: These models summarize data and describe patterns or trends without making predictions. They aim to understand the underlying structure of the data.

**Example**: **Clustering** (like K-means) is used to group customers based on buying habits, or **market basket analysis** to find associations between products purchased together.

**Distinction**:

* **Predictive models** focus on forecasting future outcomes.
* **Descriptive models** focus on explaining existing data and uncovering patterns.

**3. Describe the method of assessing a classification model's efficiency in detail. Describe the various measurement parameters.**

To assess the efficiency of a **classification model**, various metrics are used, including:

* **Accuracy**: The proportion of correct predictions (both true positives and true negatives) to the total number of predictions. Accuracy=TP+TNTP+TN+FP+FN\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
* **Precision**: The proportion of true positives among all instances predicted as positive. Precision=TPTP+FP\text{Precision} = \frac{TP}{TP + FP}
* **Recall (Sensitivity)**: The proportion of true positives among all actual positives. Recall=TPTP+FN\text{Recall} = \frac{TP}{TP + FN}
* **F1-Score**: The harmonic mean of precision and recall, providing a balance between them. F1=2×Precision×RecallPrecision+RecallF1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
* **Specificity**: The proportion of true negatives among all actual negatives. Specificity=TNTN+FP\text{Specificity} = \frac{TN}{TN + FP}
* **Kappa (Cohen’s Kappa)**: Measures agreement between predicted and actual classifications, accounting for chance. Kappa=Po−Pe1−Pe\text{Kappa} = \frac{P\_o - P\_e}{1 - P\_e}

Where PoP\_o is the observed agreement, and PeP\_e is the expected agreement by chance.

**4.**

i. **In the sense of machine learning models, what is underfitting? What is the most common reason for underfitting?**

**Underfitting** occurs when a model is too simple to capture the underlying patterns in the data, leading to poor performance on both the training and test datasets. The most common reason for underfitting is using a model that is not complex enough to model the data’s underlying relationships, such as using linear models for non-linear data.

ii. **What does it mean to overfit? When is it going to happen?**

**Overfitting** happens when a model learns not only the underlying patterns but also the noise and fluctuations in the training data, which reduces its ability to generalize to new data. It typically occurs when the model is too complex, such as a deep neural network with too many layers for a small dataset.

iii. **In the sense of model fitting, explain the bias-variance trade-off.**

The **bias-variance trade-off** refers to the balance between:

* **Bias**: Error introduced by approximating a real-world problem with a simple model (underfitting).
* **Variance**: Error introduced by the model being too sensitive to small fluctuations in the training data (overfitting).

Ideally, you want to minimize both bias and variance. A high-bias, low-variance model will underfit, while a low-bias, high-variance model will overfit. A balanced model achieves good generalization on unseen data.

**5. Is it possible to boost the efficiency of a learning model? If so, please clarify how.**

Yes, it is possible to boost the efficiency of a learning model by:

* **Tuning hyperparameters**: Optimizing settings like learning rate, number of estimators, or tree depth in decision trees.
* **Feature engineering**: Selecting or creating more relevant features that better capture the underlying patterns.
* **Ensemble methods**: Combining multiple models (e.g., bagging, boosting) to improve performance and reduce overfitting.
* **Regularization**: Adding penalties to the model to prevent overfitting (e.g., L1/L2 regularization).
* **Cross-validation**: Ensuring the model performs well on different subsets of the data to avoid overfitting.

**6. How would you rate an unsupervised learning model's success? What are the most common success indicators for an unsupervised learning model?**

To evaluate an **unsupervised learning model**:

* **Clustering quality**: Metrics like **Silhouette Score**, which measures how similar points within a cluster are to each other, and how different they are from points in other clusters.
* **Visual inspection**: Using techniques like PCA or t-SNE to visualize how well the model has separated or grouped the data.
* **Homogeneity and Completeness**: Measures how well the model's clusters match the true classes (if known) and whether all points in a class are assigned to the same cluster.

**7. Is it possible to use a classification model for numerical data or a regression model for categorical data with a classification model? Explain your answer.**

* **Classification Model for Numerical Data**: Yes, a classification model can be used for numerical data if the problem is framed as a classification task (e.g., classifying numbers into different categories).
* **Regression Model for Categorical Data**: No, regression models are designed to predict continuous output, not categories. However, categorical data can be encoded into numerical values, and a classification model should be used.

**8. Describe the predictive modeling method for numerical values. What distinguishes it from categorical predictive modeling?**

For **numerical predictive modeling**, regression techniques are used to predict a continuous target variable based on input features. Common algorithms include **linear regression**, **ridge regression**, or **support vector regression**.

In **categorical predictive modeling**, the goal is to classify data into distinct categories or classes. Common techniques include **logistic regression**, **decision trees**, or **random forests**. The key distinction is that numerical prediction involves continuous values, while categorical prediction involves discrete class labels.

**9. The following data were collected when using a classification model to predict the malignancy of a group of patients' tumors:**

* **Accurate estimates**: 15 cancerous, 75 benign
* **Wrong predictions**: 3 cancerous, 7 benign

**Determine the model's error rate, Kappa value, sensitivity, precision, and F-measure:**

* **Error Rate**:

Error Rate=FP+FNTP+TN+FP+FN=3+715+75+3+7=10100=0.1\text{Error Rate} = \frac{FP + FN}{TP + TN + FP + FN} = \frac{3 + 7}{15 + 75 + 3 + 7} = \frac{10}{100} = 0.1

* **Kappa Value**:
  + Observed agreement (P\_o) = 15+75100=0.9\frac{15 + 75}{100} = 0.9
  + Expected agreement (P\_e) = (15+3)(15+7)1002+(75+7)(75+3)1002=0.85\frac{(15 + 3)(15 + 7)}{100^2} + \frac{(75 + 7)(75 + 3)}{100^2} = 0.85

Kappa=0.9−0.851−0.85=0.050.15=0.33\text{Kappa} = \frac{0.9 - 0.85}{1 - 0.85} = \frac{0.05}{0.15} = 0.33

* **Sensitivity** (Recall):

Sensitivity=TPTP+FN=1515+3=0.83\text{Sensitivity} = \frac{TP}{TP + FN} = \frac{15}{15 + 3} = 0.83

* **Precision**:

Precision=TPTP+FP=1515+3=0.83\text{Precision} = \frac{TP}{TP + FP} = \frac{15}{15 + 3} = 0.83

* **F-Measure**:

F1=2×Precision×RecallPrecision+Recall=2×0.83×0.830.83+0.83=0.83F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0.83 \times 0.83}{0.83 + 0.83} = 0.83

**10. Make quick notes on:**

1. **The process of holding out**: Holding out refers to reserving a portion of the dataset for testing the model, while the remaining data is used for training. This technique helps to estimate the model’s generalization ability.
2. **Cross-validation by tenfold**: In **10-fold cross-validation**, the dataset is divided into 10 subsets. The model is trained on 9 of the subsets and tested on the remaining one. This is repeated for all subsets, and the performance is averaged to give an estimate of model performance.
3. **Adjusting the parameters**: Hyperparameter tuning involves adjusting the settings of a machine learning model (like the learning rate, number of trees, or depth of a tree) to improve performance, typically using methods like grid search or random search

.

**11. Define the following terms:**

1. **Purity vs. Silhouette width**:
   * **Purity**: A measure of how homogenous clusters are, calculated as the proportion of the dominant class within each cluster.
   * **Silhouette Width**: A metric used to assess how similar an object is to its own cluster compared to other clusters. A higher value indicates better-defined clusters.
2. **Boosting vs. Bagging**:
   * **Boosting**: A technique where multiple models are trained sequentially, with each new model correcting the errors of the previous ones (e.g., **AdaBoost**).
   * **Bagging**: An ensemble method where multiple models are trained in parallel, each using a different random subset of the data (e.g., **Random Forests**).
3. **The eager learner vs. the lazy learner**:
   * **Eager Learner**: A model that generalizes during training, creating a full model in advance (e.g., **decision trees**, **SVM**).
   * **Lazy Learner**: A model that does not generalize during training but makes predictions by comparing new instances to the training data at the time of prediction (e.g., **K-NN**).