Q1.What Are the Different Types of Machine Learning?

Relation →

9 → 0

Supervised Learning:

In supervised learning, the algorithm is trained on a labeled dataset, where each input example is paired with its corresponding output label. The goal is for the model to learn the mapping between input features and output labels so that it can make accurate predictions on new, unseen data.

Common Tasks:

Classification: Predicting categories or classes for new data points. For example, classifying emails as spam or not spam, or predicting whether a tumor is benign or malignant.

Regression: Predicting continuous numerical values. For instance, predicting house prices based on features like location, size, and number of bedrooms.

Unsupervised Learning:

Unsupervised learning involves training algorithms on datasets without labeled responses. Instead of learning from explicit feedback, the algorithm tries to identify patterns, relationships, or structures within the data.

Common Tasks:

Clustering: Grouping similar data points together based on some measure of similarity. For example, clustering customers into segments based on their purchasing behavior.

Dimensionality Reduction: Reducing the number of input variables or features while preserving the essential information. This is useful for visualizing high-dimensional data or speeding up subsequent computations.

Reinforcement Learning:

Reinforcement learning is about training agents to interact with an environment and learn from the consequences of their actions. The agent receives feedback in the form of rewards or penalties based on its actions, which guides its learning process.

Common Tasks:

Markov Decision Processes (MDPs): Reinforcement learning problems are often formulated as MDPs, where the agent takes actions in states and receives rewards accordingly.

Q2. What is Overfitting, and How Can You Avoid It?

The Overfitting is a situation that occurs when a model learns the training set too well, taking up random fluctuations in the training data as concepts. These impact the model's ability to generalize and don't apply to new data.

When a model is given the training data, it shows 100 percent accuracy—technically a slight loss. But, when we use the test data, there may be an error and low efficiency. This condition is known as overfitting.

There are multiple ways of avoiding overfitting, such as:

Penality ⇒ Lasso, Ridge

Regularization: It involves a cost term for the features involved with the objective function

Making a simple model :With lesser variables and parameters, the variance can be reduced

Cross-validation methods like k-folds can also be used If some model parameters are likely to cause overfitting, techniques for regularization like LASSO can be used that penalize these parameters

Q3. What is 'training Set' and 'test Set' in a Machine Learning Model? How Much Data Will You Allocate for Your Training, Validation, and Test Sets?

Training Set:

The training set is a subset of the dataset used to train the machine learning model. It consists of input data paired with the corresponding correct output labels (in supervised learning). The model learns from this data by adjusting its parameters or weights through optimization algorithms (such as gradient descent) to minimize the error between its predictions and the actual labels.

Validation Set:

The validation set is used to tune hyperparameters and evaluate the performance of the model during training. It helps in preventing overfitting by providing an unbiased evaluation of the model's performance on data that it hasn't seen during training. The validation set is also used for early stopping, where training is halted when the model's performance on the validation set stops improving or starts deteriorating.

Test Set:

The test set is used to evaluate the final performance of the trained model after it has been trained and validated. It provides an unbiased estimate of the model's performance on unseen data. The test set should ideally reflect the same distribution as the training and validation sets to ensure that the model generalizes well to new, unseen data.

Data Allocation:

1000
Training – Train n. ⇒ Book
Test → Test ⇒ Testing
Vali → Penny → sample Test

The allocation of data to the training, validation, and test sets depends on various factors, including the size of the dataset, the complexity of the problem, and the available computational resources. However, some common practices and guidelines are followed:

Training Set: Typically, the majority of the data is allocated to the training set, often around 60% to 80% of the total dataset. A larger training set allows the model to learn more effectively and generalize better to unseen data.

Validation Set: The validation set is usually smaller than the training set, typically around 10% to 20% of the total dataset. This subset is used for tuning hyperparameters and monitoring the model's performance during training.

Test Set: The test set is also smaller compared to the training set and is generally around 10% to 20% of the total dataset. It is kept completely separate from the training and validation sets until the final evaluation to ensure an unbiased estimate of the model's performance.

Q4. How Do You Handle Missing or Corrupted Data in a Dataset?

Remove Rows or Columns:

If the missing values are very few and removing them won't significantly affect the dataset's integrity, you can simply delete rows or columns containing missing values.

Mean/Median/Mode Imputation:

Replace missing values with the mean (average), median (middle value), or mode (most frequent value) of the respective feature. This method is straightforward and preserves the overall distribution of the data.

Forward Fill/Backward Fill:

Fill missing values with the value from the previous or next non-missing observation along the same feature. This is useful for time-series data where missing values are often consecutive.

Interpolation:

Use interpolation techniques such as linear interpolation or spline interpolation to estimate missing values based on neighboring data points. Interpolation is particularly useful for ordered or time-series data.

Imputation Models:

Train machine learning models (e.g., k-Nearest Neighbors, Random Forests, etc.) to predict missing values based on other features in the dataset. The model learns patterns from the complete data to impute missing values.

Flagging and Encoding:

Create an additional binary feature indicating whether a value is missing or not (flagging). Then, use encoding techniques such as mean imputation or model-based imputation for the missing values in that feature.

Q5. What Are the Three Stages of Building a Model in Machine Learning?

The three stages of building a machine learning model are:

Model Building Choose a suitable algorithm for the model and train it according to the requirement

Model Testing . Check the accuracy of the model through the test data

Applying the Model Make the required changes after testing and use the final model for real-time projects

Q6. What Are the Differences Between Machine Learning and Deep Learning?

Machine Learning:

Enables machines to take decisions on their own, based on past data

It needs only a small amount of data for training

Works well on the low-end system, so you don't need large machines

Most features need to be identified in advance and manually coded

The problem is divided into two parts and solved individually and then combined

Deep Learning :

Enables machines to take decisions with the help of artificial neural networks

It needs a large amount of training data

Needs high-end machines because it requires a lot of computing power

The machine learns the features from the data it is provided

The problem is solved in an end-to-end manner

Q7. What Are the Applications of Supervised Machine Learning in Modern Businesses?

Email Spam Detection

Here we train the model using historical data that consists of emails categorized as spam or not spam. This labeled information is fed as input to the model.

Healthcare Diagnosis

By providing images regarding a disease, a model can be trained to detect if a person is suffering from the disease or not.

Sentiment Analysis

This refers to the process of using algorithms to mine documents and determine whether they're positive, neutral, or negative in sentiment.

Fraud Detection

By training the model to identify suspicious patterns, we can detect instances of possible fraud.

Q8.What is Semi-supervised Machine Learning?

Supervised learning uses data that is completely labeled, whereas unsupervised learning uses no training data.

In the case of semi-supervised learning, the training data contains a small amount of labeled data and a large amount of unlabeled data.

Q9. What is the Difference Between Inductive Machine Learning and Deductive Machine Learning?

Aspect	Inductive Learning	Deductive Learning
Start with	Observations	A theory
Conclude with	A theory	Observations
Learning approach	Bottom-up	Top-down
Data need	Many specific examples	Established general rules
Generalization	Good at finding patterns	Excels at precise predictions
Handling Noise	Can be affected by noise	More robust with solid rules
Interpretation	Less clear due to data patterns	Easier to interpret, explicit rules
Thinking Skills	Creative, critical, inductive	Analytical, deductive
Applications	Real-life problem-solving	Abstract concept application
Reasoning	Goes from specifics to general	Goes from general to specific

Rules or [illegible] to draw conclusion

→ Rule based system
Expert system

→ Apply knowledge to solve problem.

Q10 .Compare K-means and KNN Algorithms.

Type of learning .	Supervised learning	Unsupervised learning
Task	Classification and regression	Clustering
Parameter	K, the number of nearest neighbors	K, the number of clusters
Input	Labeled data	Unlabeled data
Output .	Prediction or estimation of output variable based on k nearest neighbors .	Grouping of similar data points in k

Q11. What is 'naive' in the Naive Bayes Classifier?

The classifier is called 'naive' because it makes assumptions that may or may not turn out to be correct.

The algorithm assumes that the presence of one feature of a class is not related to the presence of any other feature (absolute independence of features), given the class variable.

For instance, a fruit may be considered to be a cherry if it is red in color and round in shape, regardless of other features. This assumption may or may not be right (as an apple also matches the description).

Q12. How Will You Know Which Machine Learning Algorithm to Choose for Your Classification Problem?

While there is no fixed rule to choose an algorithm for a classification problem, you can follow these guidelines:

If accuracy is a concern, test different algorithms and cross-validate them

If the training dataset is small, use models that have low variance and high bias

If the training dataset is large, use models that have high variance and little bias

Q13 . When Will You Use Classification over Regression?

Classification is used when your target is categorical, while regression is used when your target variable is continuous. Both classification and regression belong to the category of supervised machine learning algorithms.

Examples of classification problems include:

Predicting yes or no

Estimating gender

Breed of an animal

Type of color

Examples of regression problems include:

Estimating sales and price of a product

Predicting the score of a team 2024 → ipl →

Predicting the amount of rainfall

Q14 . Considering a Long List of Machine Learning Algorithms, given a Data Set, How Do You Decide Which One to Use?

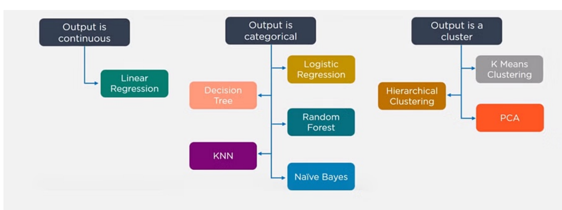
There is no master algorithm for all situations. Choosing an algorithm depends on the following questions:

How much data do you have, and is it continuous or categorical?

Is the problem related to classification, association, clustering, or regression?

Predefined variables (labeled), unlabeled, or mix?

What is the goal?



Q15.What is Bias and Variance in a Machine Learning Model?

Bias

Bias in a machine learning model occurs when the predicted values are further from the actual values. Low bias indicates a model where the prediction values are very close to the actual ones.

Underfitting: High bias can cause an algorithm to miss the relevant relations between features and target outputs.

Low Bias, Low Variance → Underfitting

Variance

Variance refers to the amount the target model will change when trained with different training data. For a good model, the variance should be minimized.

Overfitting: High variance can cause an algorithm to model the random noise in the training data rather than the intended outputs.

High variance, Low Bias = overfitting

Q16. What is the Trade-off Between Bias and Variance?

High Bias-Low Variance: Models with high bias and low variance tend to be simple and less flexible. They may not capture all the nuances in the data but are less affected by fluctuations in the training data. Examples include linear models or models with few parameters.

Low Bias-High Variance: Models with low bias and high variance tend to be more complex and flexible. They can capture intricate patterns in the data but are prone to overfitting, meaning they may perform well on the training data but generalize poorly to unseen data. Examples include decision trees with no pruning or deep neural networks.

The bias-variance trade-off in machine learning is about finding the right balance between simplicity and flexibility in a model. Too simple, and the model may miss important patterns (high bias); too complex, and it may overfit to noise in the data (high variance). The goal is to strike a balance that generalizes well to new data while capturing the underlying trends effectively.

Q17 . Achieving the balance between bias and variance involves several techniques:

Model Complexity: Start with a simple model and gradually increase complexity as needed. This helps avoid overfitting initially.

Cross-Validation: Use techniques like k-fold cross-validation to evaluate model performance on different subsets of the data. This helps in understanding how well the model generalizes to unseen data.

Regularization: Introduce regularization techniques like L1 (Lasso) or L2 (Ridge) regularization to penalize overly complex models, discouraging them from fitting to noise.

Feature Selection: Select only the most relevant features to reduce the complexity of the model and prevent overfitting.

Ensemble Methods: Combine predictions from multiple models to reduce variance. Techniques like bagging (bootstrap aggregating) and boosting help in building robust models.

Hyperparameter Tuning: Tune model hyperparameters using techniques like grid search or random search to find the optimal settings that balance bias and variance.

Bias-Variance Decomposition: Understand the sources of error in the model by decomposing the overall error into bias and variance components. This helps in diagnosing model performance issues.

Q18.Define Precision and Recall.

Precision

Precision is the ratio of several events you can correctly recall to the total number of events you recall (mix of correct and wrong recalls).

Precision = (True Positive) / (True Positive + False Positive)

Recall

A recall is the ratio of the number of events you can recall the number of total events.

Recall = (True Positive) / (True Positive + False Negative)

Q19. What do you understand by Type I vs Type II error?

2023 → [illegible] — [illegible]

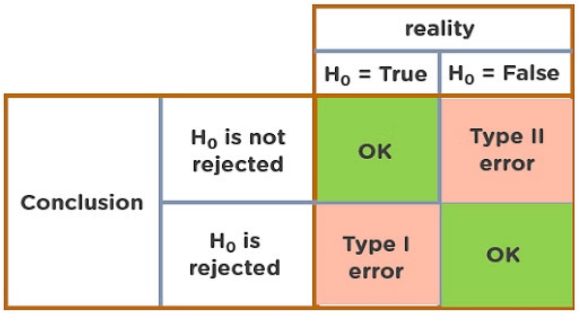
[illegible]

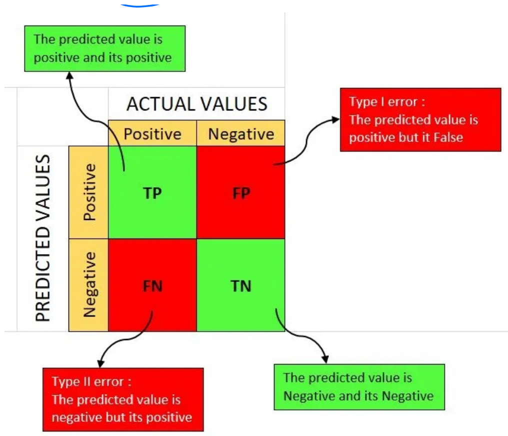
Type I Error: Type I error occurs when the null hypothesis is true and we reject it.

Type II Error: Type II error occurs when the null hypothesis is false and we accept it.

Model — Conclusion → CSK — False

Reality = $\boxed{\text{CSK}}$ — True





Q21 .What is Cross-Validation?

Cross-validation is a technique used to evaluate the performance of a machine learning model by partitioning the dataset into subsets, training the model on a portion of the data, and then evaluating it on the remaining unseen data. This process is repeated multiple times, with different partitions of the data, and the performance metrics are averaged across all iterations.

The main steps of cross-validation are as follows:

Splitting the Data: The dataset is divided into k subsets of approximately equal size. One of these subsets is held out as the validation set, while the remaining k-1 subsets are used for training.

Training and Validation: The model is trained on the k-1 subsets and evaluated on the validation set. This process is repeated k times, each time using a different subset as the validation set.

Performance Evaluation: The performance metrics, such as accuracy, precision, recall, or mean squared error, are calculated for each iteration of the cross-validation process. These metrics are then averaged to obtain a more robust estimate of the model's performance.

Q22. What are the assumptions you need to take before starting with linear regression?

There are primarily 5 assumptions for a Linear Regression model:

Multivariate normality

No auto-correlation

Homoscedasticity

Linear relationship

No or little multicollinearity

Regularization

↳ Penality

Q23. What is the difference between Lasso and Ridge regression?

The key difference is in how they assign penalties to the coefficients:

Ridge Regression:

Performs L2 regularization, i.e., adds penalty equivalent to the square of the magnitude of coefficients

Minimization objective =

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

LS Obj + α * (sum of square of coefficients)

Lasso Regression:

Performs L1 regularization, i.e., adds penalty equivalent to the absolute value of the magnitude of coefficients

Minimization objective =

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|.$$

LS Obj + α * (sum of the absolute value of coefficients)

Q24. Key Difference Between Ridge Regression and LASSO Regression

Feature Selection: Lasso can set coefficients to zero, effectively performing feature selection, while Ridge can only shrink coefficients close to zero.

Bias-Variance Tradeoff: Both methods introduce bias into estimates but reduce variance, potentially leading to better overall model predictions.

Regularization Technique: Ridge uses L2 regularization (squares of coefficients), and Lasso uses L1 regularization (absolute values of coefficients).

Outlier :- Lasso

Ridge → Feature selection

Predictive Performance: The choice between Ridge and Lasso depends on the data and the problem. Ridge tends to perform better for many significant predictors, while Lasso is more effective when only a few predictors are actually significant.

Q25.What do you understand by the F1 score?

Accura

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Harmonic Mean Preciso & Recall

$$\boxed{F_1 = \frac{2 \times P \times R}{P + R}}$$

The F1 score is a metric commonly used to evaluate the performance of a classification model. It considers both the precision and recall of the model to provide a single numerical value that summarizes its effectiveness.

Precision: Precision measures the proportion of true positive predictions out of all positive predictions made by the model.

Recall (or Sensitivity): Recall measures the proportion of true positive predictions out of all actual positive instances in the dataset

Supervised → Target Value Present

Unsupervised → Target Value Not present

Reinforcement → Objective Based

Q26 .What are the different types of machine learning?

Algorithms →

$\boxed{\text{Regression}}$

1) Linear Regression
2) Decision Tree
3) Random Forest
4) Xgboost

Classification →

1) Logistic Regression
2) Decision Tree
3) Random Forest
4) KNN
5) SVM

Clustering
- K-means
- Hierarchical
- DBSCAN
- Optics

Dimension Reduction
- PCA
- ICA
- Factor Analysis

Association
- Apriori
- FP Growth

Q27 . What are the different types of Unsupervised machine learning?

+

Q28. How do you know which machine learning algorithm should be used?

Regression: Predicting a Number (Linear Regression, Decision Trees, Random Forests, XGboost, Adaboost)

Classification: Predicting a Category (Logistic Regression, Decision Trees, Random Forests, XGboost, KNN, SVM)

Clustering: Grouping similar rows (K-Means, Hierarchical clustering, DBSCAN, OPTICS)

Dimension reduction: Reducing the number of variables in data (PCA, ICA, T-SNA, UMAP)

Association: Finding out which products sell together (Apriori, Eclat, FP-Growth)

Q29 .Can you explain the concept of overfitting in supervised learning?

Overfitting is a common problem in supervised machine learning where a model learns to capture the noise and random fluctuations in the training data instead of the underlying pattern or trend. This leads to poor generalization performance, meaning the model performs well on the training data but fails to generalize to unseen data.

Issues:

Overfitting occurs when the model is too complex for the available data, capturing noise instead of patterns.

It happens when the model fits the training data too closely, including noise and random fluctuations.

High variance accompanies overfitting, indicating the model's predictions fluctuate widely with data changes.

Signs of overfitting include low training error but high validation error, poor performance on new data, and significant gaps between training and validation metrics.

Q30 .prevent overfiting ?

Regularization: Introducing penalties on the model parameters to prevent them from becoming too large.

Cross-validation: Evaluating the model's performance on multiple subsets of the data to ensure it generalizes well.

$$\text{Accracy} = \frac{\text{No of Corret predition}}{\text{Total No of Preditions}}$$

Feature selection: Selecting only the most relevant features to reduce the complexity of the model.

Using simpler models: Choosing a simpler model architecture that is less prone to overfitting.

Q31 . What evaluation metrics would you use to assess the performance of a classification model?

Accuracy: Accuracy measures the proportion of correctly classified instances out of all instances. It is calculated as the ratio of the number of correct predictions to the total number of predictions made .

$$\text{Precasion} = \frac{TP}{TP + FP}$$

Precision: Precision measures the proportion of true positive predictions out of all positive predictions made by the model. It is calculated as the ratio of true positives to the sum of true

$$\text{Recall} = \frac{TP}{\quad}$$

positives and false positives. $TP + FN$

Recall (Sensitivity): Recall measures the proportion of true positive predictions out of all actual positive instances in the dataset. It is calculated as the ratio of true positives to the sum of true positives and false negatives.

$F_1 = \frac{2 \times P \times R}{P + R}$

F1 Score: The F1 score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance. It is calculated as:

$$\frac{2 \times P \times R}{P + R}$$

ROC Curve and AUC: The Receiver Operating Characteristic (ROC) curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The Area Under the ROC Curve (AUC) provides a single scalar value representing the model's ability to discriminate between positive and negative instances.

Confusion Matrix: A confusion matrix provides a tabular summary of the model's predictions versus the actual class labels, showing true positives, true negatives, false positives, and false negatives.

Q33.What are some common techniques for feature selection in supervised learning?

Filter Methods: Evaluate feature relevance before model training based on statistical properties or correlation with the target variable. Techniques include:

Correlation Analysis

Chi-Square Test

Information Gain

Variance Thresholding

Wrapper Methods: Use model performance as a criterion for feature selection. Techniques include:

Forward Selection

Backward Elimination

Recursive Feature Elimination (RFE)

Embedded Methods: Integrate feature selection into the model training process.

Lasso Regression

Tree-based Feature Importance

Dimensionality Reduction Techniques: Transform the feature space into a lower-dimensional space while preserving important information.

Principal Component Analysis (PCA)

Linear Discriminant Analysis (LDA)

Q34.How does regularization help prevent overfitting in supervised learning models?

Regularization and Overfitting:

- Regularization techniques introduce additional constraints or penalties on the model parameters during training to prevent overfitting.
- By penalizing large parameter values, regularization discourages the model from fitting the noise or random fluctuations in the training data.
- Regularization helps control the complexity of the model, ensuring that it generalizes well to unseen data by striking a balance between bias and variance.
- Common regularization techniques include L1 (Lasso) and L2 (Ridge) regularization, which add penalties to the absolute values and squared values of the model parameters, respectively.

Q35 .Can you describe the process of cross-validation and its importance in supervised learning?

Cross-Validation:

1. Cross-validation is a resampling technique used to evaluate the performance of a machine learning model on unseen data.
2. The process involves dividing the dataset into multiple subsets or folds, training the model on a portion of the data, and evaluating it on the remaining unseen data.
3. This process is repeated multiple times, with different partitions of the data, and the performance

metrics are averaged across all iterations to provide a more robust estimate of the model's performance.

4. Common types of cross-validation include k-fold cross-validation, leave-one-out cross-validation, and stratified cross-validation, each with its own variations and applications.
5. Cross-validation is important in supervised learning because it helps assess how well the model generalizes to new, unseen data and provides insights into its stability and reliability.
6. It helps detect issues like overfitting or underfitting by evaluating the model's performance on multiple subsets of the data, enabling the selection of the best-performing model and hyperparameters.

Q35.What is the purpose of ensemble learning in supervised machine learning?

The purpose of ensemble learning in supervised machine learning is to improve predictive performance by combining the predictions of multiple individual models. Ensemble learning techniques leverage the diversity among these models to make more accurate and robust predictions compared to any single model alone.

Key purposes and benefits of ensemble learning include:

Improved Accuracy: Ensemble methods often achieve higher predictive accuracy compared to individual models by leveraging the wisdom of crowds. By combining multiple models, ensemble learning reduces the risk of selecting a suboptimal model and can better capture the underlying patterns in the data.

Robustness: Ensemble methods are more robust to noisy data and outliers because they aggregate predictions from multiple models. They tend to be less susceptible to overfitting, as errors made by individual models may cancel out when combined.

Reduction of Bias and Variance: Ensemble learning can help strike a balance between bias and variance, leading to more stable predictions. For example, combining models with high bias and low variance (e.g., decision trees) with models with low bias and high variance (e.g., neural networks) can result in an ensemble with lower overall bias and variance.

Model Interpretability: Ensemble methods can sometimes offer better interpretability than complex individual models. For example, in bagging methods like Random Forests, feature importance can be derived from aggregating feature importance scores across multiple trees.

Versatility: Ensemble learning is versatile and can be applied to various types of models and learning tasks. It can be used with classification, regression, and even unsupervised learning tasks.

Q35.What is the purpose of ensemble learning in supervised machine learning?

The purpose of ensemble learning in supervised machine learning is to improve predictive performance by combining the predictions of multiple individual models. Ensemble learning techniques leverage the diversity among these models to make more accurate and robust predictions compared to any single model alone.

Key purposes and benefits of ensemble learning include:

Improved Accuracy: Ensemble methods often achieve higher predictive accuracy compared to individual models by leveraging the wisdom of crowds. By combining multiple models, ensemble learning reduces the risk of selecting a suboptimal model and can better capture the underlying patterns in the data.

Robustness: Ensemble methods are more robust to noisy data and outliers because they aggregate predictions from multiple models. They tend to be less susceptible to overfitting, as errors made by individual models may cancel out when combined.

Reduction of Bias and Variance: Ensemble learning can help strike a balance between bias and variance, leading to more stable predictions. For example, combining models with high bias and low variance (e.g., decision trees) with models with low bias and high variance (e.g., neural networks) can result in an ensemble with lower overall bias and variance.

Model Interpretability: Ensemble methods can sometimes offer better interpretability than complex individual models. For example, in bagging methods like Random Forests, feature importance can be derived from aggregating feature importance scores across multiple trees.

Versatility: Ensemble learning is versatile and can be applied to various types of models and learning tasks. It can be used with classification, regression, and even unsupervised learning tasks.

Q36 .Common ensemble learning techniques include:

Bagging (Bootstrap Aggregating): Constructs multiple models independently and combines their predictions through averaging or voting. Examples include Random Forests.

Boosting: Builds models sequentially, where each subsequent model focuses on correcting the errors of the previous model. Examples include AdaBoost, Gradient Boosting Machines (GBM), and XGBoost.

Stacking (Stacked Generalization): Combines predictions from multiple models using a meta-model, which learns to weigh the predictions of base models.

Q37. How do you choose between different supervised learning algorithms for a given problem?How do you choose between different supervised learning algorithms for a given problem?

For Linear Relationships: Linear regression, Logistic regression, Linear Support Vector Machines (SVM).

For Non-linear Relationships: Decision Trees, Random Forests, Gradient Boosting Machines (GBM), Neural Networks.

For High-dimensional Data: Regularized regression (e.g., Lasso, Ridge), Support Vector Machines (SVM), Ensemble methods (e.g., Random Forests, Gradient Boosting).

For Text Data: Naive Bayes, Support Vector Machines (SVM), Recurrent Neural Networks (RNNs), Transformer models (e.g., BERT).

Q38. What are the assumptions of Linear Regression?

Linear Relationship between input and output – Linear relationship means if one increases, the other should also increase or vice-versa. Linear regression assumes that the input and output are linear dependents on each other.

No Multicollinearity – Multicollinearity means that all the input columns you have in data should not be highly related. For example, if we have X1, X2, and X3 as input columns in data and if by changing X1 there are changes observed in X2, then it is the scenario of multicollinearity. For example, if by changing X1, we get output keeping that X2 and X3 are constant, but when X1 is correlated to X2, then on changing X1, X2 will also change, so we will not get proper output. That's why multicollinearity is a problem.

Normality of Residual – When we predict new data points and calculate error or residual (actual – predicted), then on plotting, it should be normally distributed across the mean. To know this, you can directly plot the KDE or QQ plot. First, you have to calculate the residual for every test point, and using the seaborn library; you can plot a distribution plot. The second way is to directly Plot a QQ plot where all points should be closer to the line.

Homoscedasticity – Home means same, and scedasticity means to spread or scatter. So this means having the same scatter. According to this assumption, when you plot residual, then the spread should be equal. If it is not equal, then it is known as Heteroscedasticity. To calculate this, you keep prediction on X-axis and residual on Y-axis. The scatter plot should be uniform.

No Autocorrelation of Errors – If you plot all residual errors, then there should not be a particular pattern.

Q38 .What are the key differences between logistic regression and linear regression ?

Problem Type:

Linear Regression: Linear regression is used for predicting continuous numeric outcomes. It models the relationship between the independent variables (features) and the continuous dependent variable (target) using a linear equation.

Logistic Regression: Logistic regression is used for predicting binary outcomes or probabilities. It models the probability that a given input belongs to a particular class using a logistic function, which restricts the output to be between 0 and 1.

Output:

Linear Regression: The output of linear regression is a continuous numeric value that represents the predicted outcome.

Logistic Regression: The output of logistic regression is a probability value between 0 and 1, which can be interpreted as the likelihood of the input belonging to a certain class. It is often used to classify inputs into one of two classes based on a threshold (e.g., 0.5).

Model Representation:

Linear Regression: In linear regression, the relationship between the input features and the target variable is represented by a straight line in a multidimensional space (hyperplane).

Logistic Regression: In logistic regression, the relationship between the input features and the log-odds of the target variable is represented by a sigmoid (logistic) function, which transforms the output to the range [0, 1].

Loss Function:

Linear Regression: The loss function used in linear regression is typically the Mean Squared Error (MSE) or Mean Absolute Error (MAE), which measures the difference between the predicted and actual values.

Logistic Regression: The loss function used in logistic regression is the Binary Cross-Entropy (also known as Log Loss), which measures the difference between the predicted probabilities and the true binary labels.

Applications:

Linear Regression: Common applications of linear regression include predicting house prices, sales forecasting, and estimating the relationship between variables.

Logistic Regression: Common applications of logistic regression include binary classification tasks such as spam detection, disease diagnosis, and customer churn prediction.

Q40. How do support vector machines (SVMs) work in supervised learning? What is the kernel trick?

Basic Concept of SVM:

SVMs work by finding the optimal hyperplane that separates data points of different classes in a high-dimensional feature space.

The goal of SVMs is to maximize the margin, which is the distance between the hyperplane and the nearest data points (support vectors) of each class.

SVMs can handle both linearly separable and non-linearly separable data by mapping the input features into a higher-dimensional space where the classes become separable.

Linear Separable Data:

For linearly separable data, SVMs find the hyperplane that maximizes the margin while ensuring that all data points are correctly classified.

The hyperplane is determined by solving an optimization problem that involves minimizing the norm of the weight vector subject to the constraint that all data points are correctly classified.

Non-linearly Separable Data:

For non-linearly separable data, SVMs use a technique called the kernel trick to map the input features into a higher-dimensional space where the classes become separable.

The kernel trick allows SVMs to implicitly compute the dot product between the mapped feature vectors in the higher-dimensional space without explicitly transforming the data.

Kernel Trick:

The kernel trick is a mathematical technique that allows SVMs to operate in a high-dimensional feature space without explicitly computing the transformed feature vectors.

Instead of explicitly transforming the input features, the kernel function computes the dot product between the input feature vectors in the higher-dimensional space.

Q41,. Bias and variance ?

Bias measures how well a model approximates the true relationship between features and target variable. High bias can cause underfitting.

Variance measures how much the model's predictions vary across different training datasets. High variance can cause overfitting.

Q42 , Regularisation :

Regularization is a technique used in machine learning and statistics to prevent overfitting and improve the generalization performance of models. It involves adding a penalty term to the model's loss function, which discourages the model from fitting the training data too closely or from becoming too complex.

The basic idea behind regularization is to impose constraints on the model parameters during training, preventing them from taking extreme or overly complex values. This helps to ensure that the model captures the underlying patterns in the data without overfitting to noise or random fluctuations.

L1 Regularization (Lasso):

L1 regularization adds a penalty term proportional to the absolute values of the model parameters to the loss function.

This penalty encourages sparsity in the model by forcing some of the parameters to be exactly zero, effectively performing feature selection.

L1 regularization is useful when the dataset contains many irrelevant or redundant features.

L2 Regularization (Ridge):

L2 regularization adds a penalty term proportional to the squared values of the model parameters to the loss function.

This penalty encourages smaller parameter values, effectively shrinking the coefficients towards zero but rarely exactly to zero.

L2 regularization is useful for reducing the impact of multicollinearity and stabilizing the model's predictions.

Q43 . cross validation :

Cross-validation is a resampling technique used in machine learning to evaluate the performance of a model on unseen data. It is commonly used to assess how well a predictive model generalizes to new data and to estimate its performance metrics.

The basic idea behind cross-validation is to partition the available dataset into multiple subsets or folds. The model is trained on a subset of the data, called the training set, and then evaluated on the remaining subset, called the validation set. This process is repeated multiple times, with different partitions of the data, and the performance metrics are averaged across all iterations to provide a more robust estimate of the model's performance.

There are several common types of cross-validation techniques:

K-Fold Cross-Validation:

The dataset is divided into K equally sized folds.

The model is trained K times, each time using K-1 folds for training and one fold for validation.

The performance metrics are averaged across all K iterations to obtain the final evaluation.

Leave-One-Out Cross-Validation (LOOCV):

Each data point is held out once as the validation set, and the model is trained on the remaining data.

This process is repeated for each data point in the dataset.

LOOCV is computationally expensive for large datasets but provides an unbiased estimate of the model's performance.

Stratified Cross-Validation:

Ensures that each fold has a similar distribution of classes or target variable values as the original dataset.

Particularly useful for imbalanced datasets where certain classes are underrepresented.

Q44. Hyper[

Hyperparameters are configuration settings that are external to the model and are not learned from the data during the training process. These parameters control the behavior of the learning algorithm and influence the performance and complexity of the model.

Definition:

Hyperparameters are parameters that are set prior to training and remain fixed throughout the training process.

They are distinct from model parameters, which are learned from the training data during the optimization process.

Examples:

Learning rate in gradient descent-based optimization algorithms.

Regularization parameter in L1 or L2 regularization techniques.

Number of hidden layers and units in a neural network.

Depth and number of trees in a decision tree-based model.

Kernel type and kernel parameters in support vector machines (SVMs).

Tuning:

Hyperparameter tuning, also known as hyperparameter optimization, involves selecting the optimal values for hyperparameters to improve the performance of the model.

This process typically involves conducting experiments with different combinations of hyperparameters and evaluating the model's performance using techniques like cross-validation.

Automated methods such as grid search, random search, and Bayesian optimization are commonly used for hyperparameter tuning.

Impact on Model:

Hyperparameters can significantly impact the performance, complexity, and generalization ability of the model.

Proper selection of hyperparameters is crucial for achieving the desired balance between bias and variance and for building models that generalize well to unseen data.

Suboptimal hyperparameters can lead to issues such as underfitting, overfitting, or poor model performance

Q44 . Accuracy in regression model :

Mean Absolute Error (MAE):

MAE measures the average absolute difference between the predicted and actual values.

It provides a straightforward interpretation of the average prediction error.

Mean Squared Error (MSE):

MSE measures the average squared difference between the predicted and actual values.

It penalizes larger errors more heavily than smaller errors and is widely used in optimization algorithms.

Root Mean Squared Error (RMSE):

RMSE is the square root of the MSE and provides a measure of the average magnitude of the errors.

It is in the same unit as the target variable and is easier to interpret than MSE.

Coefficient of Determination (R-squared):

R-squared measures the proportion of the variance in the dependent variable that is explained by the independent variables.

It ranges from 0 to 1, with higher values indicating a better fit of the model to the data.

Mean Absolute Percentage Error (MAPE):

MAPE measures the average percentage difference between the predicted and actual values relative to the actual values.

It provides a relative measure of accuracy and is useful for comparing models across different datasets.

Adjusted R-squared:

Adjusted R-squared is a modified version of R-squared that penalizes the inclusion of unnecessary variables in the model.

It accounts for the number of predictors in the model and is more appropriate for comparing models with different numbers of variables.