General Linear Model:

Q1. What is the purpose of the General Linear Model (GLM)?
Ans- It is used to model the relationship between a dependent variable and one or more independent variables. It provides a flexible approach to analyze and understand the relationships between variables, making it widely used in various fields such as regression analysis, analysis of variance (ANOVA), and analysis of covariance (ANCOVA).

Q2. What are the key assumptions of the General Linear Model?
Ans- The key assumptions of the GLM:

1. Linearity: The GLM assumes that the relationship between the dependent variable and the independent variables is linear. This means that the effect of each independent variable on the dependent variable is additive and constant across the range of the independent variables.

2. Independence: The observations or cases in the dataset should be independent of each other. This assumption implies that there is no systematic relationship or dependency between observations. Violations of this assumption, such as autocorrelation in time series data or clustered observations, can lead to biased and inefficient parameter estimates.

3. Homoscedasticity: Homoscedasticity assumes that the variance of the errors (residuals) is constant across all levels of the independent variables. In other words, the spread of the residuals should be consistent throughout the range of the predictors. Heteroscedasticity, where the variance of the errors varies with the levels of the predictors, violates this assumption and can impact the validity of statistical tests and confidence intervals.

4. Normality: The GLM assumes that the errors or residuals follow a normal distribution. This assumption is necessary for valid hypothesis testing, confidence intervals, and model inference. Violations of normality can affect the accuracy of parameter estimates and hypothesis tests.

5. No Multicollinearity: Multicollinearity refers to a high degree of correlation between independent variables in the model. The GLM assumes that the independent variables are not perfectly correlated with each other, as this can lead to instability and difficulty in estimating the individual effects of the predictors.

6. No Endogeneity: Endogeneity occurs when there is a correlation between the error term and one or more independent variables. This violates the assumption that the errors are independent of the predictors and can lead to biased and inconsistent parameter estimates.

7. Correct Specification: The GLM assumes that the model is correctly specified, meaning that the functional form of the relationship between the variables is accurately represented in the model. Omitting relevant variables or including irrelevant variables can lead to biased estimates and incorrect inferences.

Q3. How do you interpret the coefficients in a GLM?
Ans- The coefficients in the GLM can be interpreted as:

1. Coefficient Sign:
The sign (+ or -) of the coefficient indicates the direction of the relationship between the independent variable and the dependent variable. A positive coefficient indicates a positive relationship, meaning that an increase in the independent variable is associated with an increase in the dependent variable. Conversely, a negative coefficient indicates a negative relationship, where an increase in the independent variable is associated with a decrease in the dependent variable.

2. Magnitude:
The magnitude of the coefficient reflects the size of the effect that the independent variable has on the dependent variable, all else being equal. Larger coefficient values indicate a stronger influence of the independent variable on the dependent variable. For example, if the coefficient for a variable is 0.5, it means that a one-unit increase in the independent variable is associated with a 0.5-unit increase (or decrease, depending on the sign) in the dependent variable.

3. Statistical Significance:
The statistical significance of a coefficient is determined by its p-value. A low p-value (typically less than 0.05) suggests that the coefficient is statistically significant, indicating that the relationship between the independent variable and the dependent variable is unlikely to occur by chance. On the other hand, a high p-value suggests that the coefficient is not statistically significant, meaning that the relationship may not be reliable.

4. Adjusted vs. Unadjusted Coefficients:
In some cases, models with multiple independent variables may include adjusted coefficients. These coefficients take into account the effects of other variables in the model. Adjusted coefficients provide a more accurate estimate of the relationship between a specific independent variable and the dependent variable, considering the influences of other predictors.

Q4. What is the difference between a univariate and multivariate GLM?
Ans- In univariate analysis on single variable is performed whereas in multivariate analysis on more than one variable is done. Purpose multivariate analysis is to find relationship among the variables.

Q5. Explain the concept of interaction effects in a GLM.
Ans-

Q6. How do you handle categorical predictors in a GLM?
Ans-Handling categorical variables in the General Linear Model (GLM) requires appropriate encoding techniques to incorporate them into the model effectively. Categorical variables represent qualitative attributes and can significantly impact the relationship with the dependent variable. Here are a few common methods for handling categorical variables in the GLM:

1. Dummy Coding (Binary Encoding):
Dummy coding, also known as binary encoding, is a widely used technique to handle categorical variables in the GLM. It involves creating binary (0/1) dummy variables for each category within the categorical variable. The reference category is represented by 0 values for all dummy variables, while the other categories are encoded with 1 for the corresponding dummy variable.

2. Effect Coding (Deviation Encoding):
Effect coding, also called deviation coding, is another encoding technique for categorical variables in the GLM. In effect coding, each category is represented by a dummy variable, similar to dummy coding. However, unlike dummy coding, the reference category has -1 values for the corresponding dummy variable, while the other categories have 0 or 1 values.

3. One-Hot Encoding:
One-hot encoding is another popular technique for handling categorical variables. It creates a separate binary variable for each category within the categorical variable. Each variable represents whether an observation belongs to a particular category (1) or not (0). One-hot encoding increases the dimensionality of the data, but it ensures that the GLM can capture the effects of each category independently.

It is important to note that the choice of encoding technique depends on the specific problem, the number of categories within the variable, and the desired interpretation of the coefficients. Additionally, in cases where there are a large number of categories, other techniques like entity embedding or feature hashing may be considered.

Q7. What is the purpose of the design matrix in a GLM?
Ans-
The design matrix, also known as the model matrix or feature matrix, is a crucial component of the General Linear Model (GLM). It is a structured representation of the independent variables in the GLM, organized in a matrix format. The design matrix serves the purpose of encoding the relationships between the independent variables and the dependent variable, allowing the GLM to estimate the coefficients and make predictions. Here's the purpose of the design matrix in the GLM:

1. Encoding Independent Variables:
The design matrix represents the independent variables in a structured manner. Each column of the matrix corresponds to a specific independent variable, and each row corresponds to an observation or data point. The design matrix encodes the values of the independent variables for each observation, allowing the GLM to incorporate them into the model.

2. Incorporating Nonlinear Relationships:
The design matrix can include transformations or interactions of the original independent variables to capture nonlinear relationships between the predictors and the dependent variable. For example, polynomial terms, logarithmic transformations, or interaction terms can be included in the design matrix to account for nonlinearities or interactions in the GLM.

3. Handling Categorical Variables:
Categorical variables need to be properly encoded to be included in the GLM. The design matrix can handle categorical variables by using dummy coding or other encoding schemes. Dummy variables are binary variables representing the categories of the original variable. By encoding categorical variables appropriately in the design matrix, the GLM can incorporate them in the model and estimate the corresponding coefficients.

4. Estimating Coefficients:
The design matrix allows the GLM to estimate the coefficients for each independent variable. By incorporating the design matrix into the GLM's estimation procedure, the model determines the relationship between the independent variables and the dependent variable, estimating the magnitude and significance of the effects of each predictor.

5. Making Predictions:
Once the GLM estimates the coefficients, the design matrix is used to make predictions for new, unseen data points. By multiplying the design matrix of the new data with the estimated coefficients, the GLM can generate predictions for the dependent variable based on the values of the independent variables.

Q8. How do you test the significance of predictors in a GLM?
Ans-

Q9. What is the difference between Type I, Type II, and Type III sums of squares in a GLM?
Ans-

Q10. Explain the concept of deviance in a GLM.
Ans-

Regression:

Q11. What is regression analysis and what is its purpose?
Ans-Regression analysis is a statistical technique used to model the relationship between a dependent variable and one or more independent variables. It aims to understand how changes in the independent variables are associated with changes in the dependent variable. Regression analysis helps in predicting and estimating the values of the dependent variable based on the values of the independent variables.

12. What is the difference between simple linear regression and multiple linear regression?
Ans- In simple linear regression there is one independent(X) and one dependent feature(Y) and the relation is linear with equation is Y=mX+c . In multiple linear regression there is more than one independent feature (X1, X2, X3….) and one dependent (Y) feature and relation is linear with the equation Y=m1X1+m2X2…+c

13. How do you interpret the R-squared value in regression?
Ans- R-squared (Coefficient of Determination) is a widely used measure to assess the goodness of fit in regression. It represents the proportion of the variance in the dependent variable that can be explained by the independent variables in the model. R-squared ranges from 0 to 1, with a higher value indicating a better fit.

14. What is the difference between correlation and regression?
Ans- Correlation only shows relation b/w 2 or more variables whereas regression is able to show cause-and-effect relation b/w dependent and independent variables via equation. Values of correlation generally lie from -1 to 1 which is not the same with regression.

15. What is the difference between the coefficients and the intercept in regression?
Ans-Coefficients represent slope of independent variables with the dependent variable and intercept represent value of dependent variable if all independent variables become zero.

16. How do you handle outliers in regression analysis?
Ans- Outliers can be handled in following few ways:
1 Removing/Deleting the outliers. It is generally done when large dataset is available and number of outliers are not significant.
2. Capping the outliers with quantiles. The outliers below the lower quantile are increased to lower quantile value and the values above the higher quantile are reduced to upper quantile value.
3.Making outlier values equal to median.

17. What is the difference between ridge regression and ordinary least squares regression?
Ans- Ridge regression is a form of linear regression that incorporates a regularization term to prevent overfitting and improve model performance. It is particularly useful when dealing with

multicollinearity among the independent variables. Ridge regression helps to shrink the coefficient estimates and mitigate the impact of multicollinearity, leading to more stable and reliable models.
Ordinary least squares regression involves a single independent variable (X) and a continuous dependent variable (Y). It models the relationship between X and Y as a straight line.

18. What is heteroscedasticity in regression and how does it affect the model?
Ans- It refers to the unequal scatter of residuals or error terms. Heteroscedasticity occurs when the variance of the errors varies with the levels of the predictors, violates this assumption and can impact the validity of statistical tests and confidence intervals.

19. How do you handle multicollinearity in regression analysis?
Ans- It is handled in following ways:

1. Variable Selection: Remove one or more correlated variables from the regression model to eliminate multicollinearity. Prioritize variables that are theoretically more relevant or have stronger relationships with the dependent variable.

2. Data Collection: Collect additional data to reduce the correlation between variables. Increasing sample size can help alleviate multicollinearity by providing a more diverse range of observations.

3. Ridge Regression: Use regularization techniques like ridge regression to mitigate multicollinearity. Ridge regression introduces a penalty term that shrinks the coefficient estimates, reducing their sensitivity to multicollinearity.

4. Principal Component Analysis (PCA): Transform the correlated variables into a set of uncorrelated principal components through techniques like PCA. The principal components can then be used as independent variables in the regression model.

Addressing multicollinearity is essential to ensure the accuracy and reliability of regression analysis. By identifying and managing multicollinearity

20. What is polynomial regression and when is it used?
Ans-Polynomial regression is an extension of linear regression that models the relationship between the independent variables and the dependent variable as a higher-degree polynomial function. It allows for capturing nonlinear relationships between the variables. For example, consider a dataset that includes information about the age of houses (X) and their corresponding sale prices (Y). Polynomial regression can be used to model how the age of a house affects its sale price and account for potential nonlinearities in the relationship.

Loss function:

21. What is a loss function and what is its purpose in machine learning?
Ans- A loss function, also known as a cost function or objective function, is a measure used to quantify the discrepancy or error between the predicted values and the true values in a machine learning or optimization problem. The choice of a suitable loss function depends on the specific task and the nature of the problem. Loss function guide the optimization process, facilitate gradient calculations, aid in model selection, and enable regularization.

22. What is the difference between a convex and non-convex loss function?
Ans- A loss function is considered convex if the second derivative is positive semi-definite, meaning that the curvature of the function is always non-negative. This property ensures that any local minimum of the loss function is also the global minimum. Convex loss functions play a crucial role in optimization problems as they guarantee the existence of a unique global minimum.
In contrast to convex loss functions, non-convex loss functions have multiple local minima and may be challenging to optimize. Non-convexity can pose challenges in finding the global minimum as optimization algorithms may get stuck in suboptimal solutions.

23. What is mean squared error (MSE) and how is it calculated?
Ans-The Mean Squared Error is a commonly used loss function for regression problems. It calculates the average of the squared differences between the predicted and true values. The goal is to minimize the MSE, which penalizes larger errors more severely

24. What is mean absolute error (MAE) and how is it calculated?
Ans- Mean Absolute Error measures the average of the absolute differences between the predicted and true values. It treats all errors equally, regardless of their magnitude, making it less sensitive to outliers compared to squared loss. Absolute loss is less influenced by extreme values and is more robust in the presence of outliers.

25. What is log loss (cross-entropy loss) and how is it calculated?
Ans- Binary Cross-Entropy loss is commonly used for binary classification problems, where the goal is to classify instances into two classes. It quantifies the difference between the predicted probabilities and the true binary labels.
Binary cross entropy compares each of the predicted probabilities to actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value. That means how close or far from the actual value. Then average ot log of these scores is taken. When the log loss value is low, it indicates a high level of accuracy in the model’s predictions.

26. How do you choose the appropriate loss function for a given problem?
Ans- Choosing an appropriate loss function for a given problem involves considering the nature of the problem, the type of learning task (regression, classification, etc.), and the specific goals or requirements of the problem. Here are some guidelines to help you choose the right loss function, along with examples:

1. Regression Problems:
For regression problems, where the goal is to predict continuous numerical values, common loss functions include:

- Mean Squared Error (MSE): This loss function calculates the average squared difference between the predicted and true values. It penalizes larger errors more severely.

Example: In predicting housing prices based on various features like square footage and number of bedrooms, MSE can be used as the loss function to measure the discrepancy between the predicted and actual prices.

- Mean Absolute Error (MAE): This loss function calculates the average absolute difference between the predicted and true values. It treats all errors equally and is less sensitive to outliers.

Example: In a regression problem predicting the age of a person based on height and weight, MAE can be used as the loss function to minimize the average absolute difference between the predicted and true ages.

2. Classification Problems:
For classification problems, where the task is to assign instances into specific classes, common loss functions include:

- Binary Cross-Entropy (Log Loss): This loss function is used for binary classification problems, where the goal is to estimate the probability of an instance belonging to a particular class. It quantifies the difference between the predicted probabilities and the true labels.

Example: In classifying emails as spam or not spam, binary cross-entropy loss can be used to compare the predicted probabilities of an email being spam or not with the true labels (0 for not spam, 1 for spam).

- Categorical Cross-Entropy: This loss function is used for multi-class classification problems, where the goal is to estimate the probability distribution across multiple classes. It measures the discrepancy between the predicted probabilities and the true class labels.

Example: In classifying images into different categories like cats, dogs, and birds, categorical cross-entropy loss can be used to measure the discrepancy between the predicted probabilities and the true class labels.

3. Imbalanced Data:
In scenarios with imbalanced datasets, where the number of instances in different classes is disproportionate, specialized loss functions can be employed to address the class imbalance. These include:

- Weighted Cross-Entropy: This loss function assigns different weights to each class to account for the imbalanced distribution. It upweights the minority class to ensure its contribution is not overwhelmed by the majority class.

Example: In fraud detection, where the number of fraudulent transactions is typically much smaller than non-fraudulent ones, weighted cross-entropy can be used to give more weight to the minority class (fraudulent transactions) and improve model performance.

4. Custom Loss Functions:
In some cases, specific problem requirements or domain knowledge may necessitate the development of custom loss functions tailored to the problem at hand. Custom loss functions allow the incorporation of specific metrics, constraints, or optimization goals into the learning process.

Example: In a recommendation system, where the goal is to optimize a ranking metric like the mean average precision (MAP), a custom loss function can be designed to directly optimize MAP during model training.

When selecting a loss function, consider factors such as the desired behavior of the model, sensitivity to outliers, class imbalance, and any specific domain considerations. Experimentation and evaluation of different loss functions can help determine which one performs best for a given problem.

27. Explain the concept of regularization in the context of loss functions.
Ans- Loss functions are often combined with regularization techniques to prevent overfitting and improve the generalization ability of models. Regularization adds a penalty term to the loss function, encouraging simpler and more robust models.

28. What is Huber loss and how does it handle outliers

Ans- Huber loss assigns less weight to observations identified as outliers. For loss values less than delta, use the MSE; for loss values greater than delta, use the MAE. This effectively combines the best of both worlds from the two loss functions. Using the MAE for larger loss values mitigates the weight that we put on outliers so that we still get a well-rounded model. At the same time we use the MSE for the smaller loss values to maintain a quadratic function near the centre.
This has the effect of magnifying the loss values as long as they are greater than 1. Once the loss for those data points dips below 1, the quadratic function down-weights them to focus the training on the higher-error data points.

29. What is quantile loss and when is it used?
Ans- The value of quantile loss depends on whether a prediction is less or greater than the true value. Model will be more critical to under-estimated errors and will predict higher values more often or vice versa. It is generally used when task is to predict quantile of variable.

30. What is the difference between squared loss and absolute loss?
Ans- Squared loss and absolute loss are two commonly used loss functions in regression problems. They measure the discrepancy or error between predicted values and true values, but they differ in terms of their properties and sensitivity to outliers. Here's an explanation of the differences between squared loss and absolute loss with examples:

Squared Loss (Mean Squared Error):
Squared loss, also known as Mean Squared Error (MSE), calculates the average of the squared differences between the predicted and true values. It penalizes larger errors more severely due to the squaring operation. The squared loss function is differentiable and continuous, which makes it well-suited for optimization algorithms that rely on gradient-based techniques.

Mathematically, the squared loss is defined as:
Loss(y, ŷ) = (1/n) * ∑(y - ŷ)^2

Example:
Consider a simple regression problem to predict house prices based on the square footage. If the true price of a house is $300,000, and the model predicts $350,000, the squared loss would be (300,000 - 350,000)^2 = 25,000,000. The larger squared difference between the predicted and true values results in a higher loss.

Absolute Loss (Mean Absolute Error):
Absolute loss, also known as Mean Absolute Error (MAE), measures the average of the absolute differences between the predicted and true values. It treats all errors equally, regardless of their magnitude, making it less sensitive to outliers compared to squared loss. Absolute loss is less influenced by extreme values and is more robust in the presence of outliers.

Mathematically, the absolute loss is defined as:
$Loss(y, \hat{y}) = (1/n) * \sum|y - \hat{y}|$

Example:
Using the same house price prediction example, if the true price of a house is $300,000 and the model predicts $350,000, the absolute loss would be |300,000 - 350,000| = 50,000. The absolute difference between the predicted and true values is directly considered without squaring it, resulting in a lower loss compared to squared loss.

Comparison:
- Sensitivity to Errors: Squared loss penalizes larger errors more severely due to the squaring operation, while absolute loss treats all errors equally, regardless of their magnitude.
- Sensitivity to Outliers: Squared loss is more sensitive to outliers because the squared differences amplify the impact of extreme values. Absolute loss is less sensitive to outliers as it only considers the absolute differences.
- Differentiability: Squared loss is differentiable, making it suitable for gradient-based optimization algorithms. Absolute loss is not differentiable at zero, which may require specialized optimization techniques.
- Robustness: Absolute loss is more robust to outliers and can provide more robust estimates in the presence of extreme values compared to squared loss.

The choice between squared loss and absolute loss depends on the specific problem, the characteristics of the data, and the desired properties of the model. Squared loss is commonly used in many regression tasks, while absolute loss is preferred when robustness to outliers is a priority or when the distribution of errors is known to be asymmetric.

Optimizer (GD):

31. What is an optimizer and what is its purpose in machine learning?
Ans- In machine learning, an optimizer is an algorithm or method used to adjust the parameters of a model in order to minimize the loss function or maximize the objective function. Optimizers play a crucial role in training machine learning models by iteratively updating the model's parameters to improve its performance. They determine the direction and magnitude of the parameter updates based on the gradients of the loss or objective function.

32. What is Gradient Descent (GD) and how does it work?
Ans- Gradient Descent is a popular optimization algorithm used in various machine learning models. It iteratively adjusts the model's parameters in the direction opposite to the gradient of the loss function. It continuously takes small steps towards the minimum of the loss function until convergence is achieved. There are different variants of gradient descent, including:

- Stochastic Gradient Descent (SGD): This variant randomly samples a subset of the training data (a batch) in each iteration, making the updates more frequent but with higher variance.

- Mini-Batch Gradient Descent: This variant combines the benefits of SGD and batch gradient descent by using a mini-batch of data for each parameter update.

33. What are the different variations of Gradient Descent?
Ans- Gradient Descent (GD) has different variations that adapt the update rule to improve convergence speed and stability. Here are three common variations of Gradient Descent:

1. Batch Gradient Descent (BGD):
Batch Gradient Descent computes the gradients using the entire training dataset in each iteration. It calculates the average gradient over all training examples and updates the parameters accordingly. BGD can be computationally expensive for large datasets, as it requires the computation of gradients for all training examples in each iteration. However, it guarantees convergence to the global minimum for convex loss functions.

Example: In linear regression, BGD updates the slope and intercept of the regression line based on the gradients calculated using all training examples in each iteration.

2. Stochastic Gradient Descent (SGD):
Stochastic Gradient Descent updates the parameters using the gradients computed for a single training example at a time. It randomly selects one instance from the training dataset and performs the parameter update. This process is repeated for a fixed number of iterations or until convergence. SGD is computationally efficient as it uses only one training example per iteration, but it introduces more noise and has higher variance compared to BGD.

Example: In training a neural network, SGD updates the weights and biases based on the gradients computed using one training sample at a time.

3. Mini-Batch Gradient Descent:
Mini-Batch Gradient Descent is a compromise between BGD and SGD. It updates the parameters using a small random subset of training examples (mini-batch) at each iteration. This approach reduces the computational burden compared to BGD while maintaining a lower variance than SGD. The mini-batch size is typically chosen to balance efficiency and stability.

Example: In training a convolutional neural network for image classification, mini-batch gradient descent updates the weights and biases using a small batch of images at each iteration.

These variations of Gradient Descent offer different trade-offs in terms of computational efficiency and convergence behavior. The choice of which variation to use depends on factors such as the dataset size, the computational resources available, and the characteristics of the optimization problem. In practice, variations like SGD and mini-batch gradient descent are often preferred for large-scale and deep learning tasks due to their efficiency, while BGD is suitable for smaller datasets or problems where convergence to the global minimum is desired.

34. What is the learning rate in GD and how do you choose an appropriate value?
Ans- Choosing an appropriate learning rate is crucial in Gradient Descent (GD) as it determines the step size for parameter updates. A learning rate that is too small may result in slow convergence, while a learning rate that is too large can lead to overshooting or instability. Here are some guidelines to help you choose a suitable learning rate in GD:

1. Grid Search:
One approach is to perform a grid search, trying out different learning rates and evaluating the performance of the model on a validation set. Start with a range of learning rates (e.g., 0.1, 0.01, 0.001) and iteratively refine the search by narrowing down the range based on the results. This approach can be time-consuming, but it provides a systematic way to find a good learning rate.

2. Learning Rate Schedules:
Instead of using a fixed learning rate throughout the training process, you can employ learning rate schedules that dynamically adjust the learning rate over time. Some commonly used learning rate schedules include:

- Step Decay: The learning rate is reduced by a factor (e.g., 0.1) at predefined epochs or after a fixed number of iterations.

- Exponential Decay: The learning rate decreases exponentially over time.

- Adaptive Learning Rates: Techniques like AdaGrad, RMSprop, and Adam automatically adapt the learning rate based on the gradients, adjusting it differently for each parameter.

These learning rate schedules can be beneficial when the loss function is initially high and requires larger updates, which can be accomplished with a higher learning rate. As training progresses and the loss function approaches the minimum, a smaller learning rate helps achieve fine-grained adjustments.

3. Momentum:
Momentum is a technique that helps overcome local minima and accelerates convergence. It introduces a "momentum" term that accumulates the gradients over time. In addition to the learning rate, you need to tune the momentum hyperparameter. Higher values of momentum (e.g., 0.9) can smooth out the update trajectory and help navigate flat regions, while lower values (e.g., 0.5) allow for more stochasticity.

4. Learning Rate Decay:
Gradually decreasing the learning rate as training progresses can help improve convergence. For example, you can reduce the learning rate by a fixed percentage after each epoch or after a certain number of iterations. This approach allows for larger updates at the beginning when the loss function is high and smaller updates as it approaches the minimum.

5. Visualization and Monitoring:
Visualizing the loss function over iterations or epochs can provide insights into the behavior of the optimization process. If the loss fluctuates drastically or fails to converge, it may indicate an inappropriate learning rate. Monitoring the learning curves can help identify if the learning rate is too high (loss oscillates or diverges) or too low (loss decreases very slowly).

It is important to note that the choice of learning rate is problem-dependent and may require some experimentation and tuning. The specific characteristics of the dataset, the model architecture, and the optimization algorithm can influence the ideal learning rate. It is advisable to start with a conservative learning rate and gradually increase or decrease it based on empirical observations and performance evaluation on a validation set.

35. How does GD handle local optima in optimization problems?
Ans- It is done by momentum. Momentum is a technique that helps overcome local optima and accelerates convergence. It introduces a "momentum" term that accumulates the gradients over time. In addition to the learning rate, you need to tune the momentum hyperparameter. Higher values of momentum (e.g., 0.9) can smooth out the update trajectory and help navigate flat regions, while lower values (e.g., 0.5) allow for more stochasticity.

36. What is Stochastic Gradient Descent (SGD) and how does it differ from GD?
Ans-Stochastic Gradient Descent updates the parameters using the gradients computed for a single training example at a time. It randomly selects one instance from the training dataset and performs the parameter update. This process is repeated for a fixed number of iterations or until convergence. SGD is computationally efficient as it uses only one training example per iteration, but it introduces more noise and has higher variance compared to BGD.

37. Explain the concept of batch size in GD and its impact on training.
Ans- Batch size indicates number of training examples considered for calculating gradient calculation and updating the parameters at a time.
There are actually three (3) cases:

- batch_size = 1 means indeed stochastic gradient descent (SGD)
- A batch_size equal to the whole of the training data is (batch) gradient descent (BGD)
- Intermediate cases (which are actually used in practice) are usually referred to as mini-batch gradient descent

Higher the batchsize more is the computational time.

38. What is the role of momentum in optimization algorithms?
Momentum is a technique that helps overcome local minima and accelerates convergence. It introduces a "momentum" term that accumulates the gradients over time. In addition to the learning rate, you need to tune the momentum hyperparameter. Higher values of momentum

(e.g., 0.9) can smooth out the update trajectory and help navigate flat regions, while lower values (e.g., 0.5) allow for more stochasticity

39. What is the difference between batch GD, mini-batch GD, and SGD?
Ans- Batch Gradient Descent computes the gradients using the entire training dataset in each iteration. It calculates the average gradient over all training examples and updates the parameters accordingly. BGD can be computationally expensive for large datasets, as it requires the computation of gradients for all training examples in each iteration. However, it guarantees convergence to the global minimum for convex loss functions.

Stochastic Gradient Descent updates the parameters using the gradients computed for a single training example at a time. It randomly selects one instance from the training dataset and performs the parameter update. This process is repeated for a fixed number of iterations or until convergence. SGD is computationally efficient as it uses only one training example per iteration, but it introduces more noise and has higher variance compared to BGD.

Mini-Batch Gradient Descent is a compromise between BGD and SGD. It updates the parameters using a small random subset of training examples (mini-batch) at each iteration. This approach reduces the computational burden compared to BGD while maintaining a lower variance than SGD. The mini-batch size is typically chosen to balance efficiency and stability.

40. How does the learning rate affect the convergence of GD?
Ans- Choosing an appropriate learning rate is crucial in Gradient Descent (GD) as it determines the step size for parameter updates. A learning rate that is too small may result in slow convergence, while a learning rate that is too large can lead to overshooting or instability.

Regularization:

41. What is regularization and why is it used in machine learning?
Ans- Regularization is a technique used in machine learning to prevent overfitting and improve the generalization ability of a model. It introduces additional constraints or penalties to the loss function, encouraging the model to learn simpler patterns and avoid overly complex or noisy representations. Regularization helps strike a balance between fitting the training data well and avoiding overfitting, thereby improving the model's performance on unseen data.
Its key purposes are:
1 Reducing Model Complexity
2. Prevent overfitting
3. Improving Generalization
4. Feature selection

42. What is the difference between L1 and L2 regularization?
Ans- L1 regularization encourages sparsity and feature selection, setting some coefficients exactly to zero. L2 regularization promotes smaller magnitudes for all coefficients without enforcing sparsity. The choice between L1 and L2 regularization depends on the problem, the nature of the features, and the desired behavior of the model.

43. Explain the concept of ridge regression and its role in regularization.
Ans- Ridge regression or L2 regularization adds a penalty term to the loss function proportional to the square of the model's coefficients. It encourages the model to reduce the magnitude of all coefficients uniformly, effectively shrinking them towards zero without necessarily setting them exactly to zero. L2 regularization can be represented as:
Loss function + λ * $||\text{coefficients}||_2^2$

44. What is the elastic net regularization and how does it combine L1 and L2 penalties?
Ans- Elastic Net regularization combines both L1 and L2 regularization techniques. It adds a linear combination of the L1 and L2 penalty terms to the loss function, controlled by two hyperparameters: α and λ. Elastic Net can overcome some limitations of L1 and L2 regularization and provides a balance between feature selection and coefficient shrinkage.
Example:
In linear regression, Elastic Net regularization can be used when there are many features and some of them are highly correlated. It can effectively handle multicollinearity by encouraging grouping of correlated features together or selecting one feature from the group.

45. How does regularization help prevent overfitting in machine learning models?
Ans- Regularization combats overfitting, which occurs when a model performs well on the training data but fails to generalize to new, unseen data. By penalizing large parameter values or encouraging sparsity, regularization discourages the model from becoming too specialized to the training data. It encourages the model to capture the underlying patterns and avoid fitting noise or idiosyncrasies present in the training set, leading to better performance on unseen data.

46. What is early stopping and how does it relate to regularization?
Ans- Early stopping is a technique to prevent model overfitting. In this technique we stop training model if the loss function on validation data does not show any improvement. Thus by preventing overfitting early stopping in a way tries to solve purpose of regularization.

47. Explain the concept of dropout regularization in neural networks.
Ans- Dropout regularization is a technique primarily used in neural networks. It randomly drops out (sets to zero) a fraction of neurons or connections during each training iteration. Dropout prevents the network from relying too heavily on a specific subset of neurons and encourages the learning of more robust and generalizable features.
Example:

In a deep neural network, dropout regularization can be applied to intermediate layers to prevent over-reliance on certain neurons or connections. This helps reduce overfitting and improves the network's generalization performance.

48. How do you choose the regularization parameter in a model?
Ans-Selecting the regularization parameter, often denoted as λ (lambda), in a model is an important step in regularization techniques like L1 or L2 regularization. The regularization parameter controls the strength of the regularization effect, striking a balance between model complexity and the extent of regularization. Here are a few approaches to selecting the regularization parameter:

1. Grid Search:
Grid search is a commonly used technique to select the regularization parameter. It involves specifying a range of potential values for λ and evaluating the model's performance using each value. The performance metric can be measured on a validation set or using cross-validation. The regularization parameter that yields the best performance (e.g., highest accuracy, lowest mean squared error) is then selected as the optimal value.

Example:
In a linear regression problem with L2 regularization, you can set up a grid search with a range of λ values, such as [0.01, 0.1, 1, 10]. Train and evaluate the model for each λ value, and choose the one that yields the best performance on the validation set.

2. Cross-Validation:
Cross-validation is a robust technique for model evaluation and parameter selection. It involves splitting the dataset into multiple subsets or folds, training the model on different combinations of the subsets, and evaluating the model's performance. The regularization parameter can be selected based on the average performance across the different folds.

Example:
In a classification problem using logistic regression with L1 regularization, you can perform k-fold cross-validation. Vary the values of λ and evaluate the model's performance using metrics like accuracy or F1 score. Select the λ value that yields the best average performance across all folds.

3. Regularization Path:
A regularization path is a visualization of the model's performance as a function of the regularization parameter. It helps identify the trade-off between model complexity and performance. By plotting the performance metric (e.g., accuracy, mean squared error) against different λ values, you can observe how the performance changes. The regularization

parameter can be chosen based on the point where the performance stabilizes or starts to deteriorate.

Example:
In a support vector machine (SVM) with L2 regularization, you can plot the accuracy or F1 score as a function of different λ values. Observe the trend and choose the λ value where the performance is relatively stable or optimal.

4. Model-Specific Heuristics:
Some models have specific guidelines or heuristics for selecting the regularization parameter. For example, in elastic net regularization, there is an additional parameter α that controls the balance between L1 and L2 regularization. In such cases, domain knowledge or empirical observations can guide the selection of the regularization parameter.

It is important to note that the choice of the regularization parameter is problem-dependent, and there is no one-size-fits-all approach. It often requires experimentation and tuning to find the optimal value. Regularization parameter selection should be accompanied by careful evaluation and validation to ensure the chosen value improves the model's generalization performance and prevents overfitting

49. What is the difference between feature selection and regularization?
Ans- Feature selection is to optimize number of input features in model. Regularization is done to prevent overfitting and improve the generalization ability of a model. Some regularization techniques, like L1 regularization, promote sparsity in the model by driving some coefficients to exactly zero. This property can facilitate feature selection, where less relevant or redundant features are automatically ignored by the model. Feature selection through regularization can enhance model interpretability and reduce computational complexity.
Regularization is particularly important when dealing with limited or noisy data, complex models with high-dimensional feature spaces, and cases where the number of features exceeds the number of observations

50. What is the trade-off between bias and variance in regularized models?
Ans-This ideal goal of generalization in terms of bias and variance is a low bias and a low variance which is near impossible or difficult to achieve. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data.
If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if our model has large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.

SVM:

51. What is Support Vector Machines (SVM) and how does it work?
52. How does the kernel trick work in SVM?
53. What are support vectors in SVM and why are they important?
54. Explain the concept of the margin in SVM and its impact on model performance.
55. How do you handle unbalanced datasets in SVM?
56. What is the difference between linear SVM and non-linear SVM?
57. What is the role of C-parameter in SVM and how does it affect the decision boundary?
58. Explain the concept of slack variables in SVM.
59. What is the difference between hard margin and soft margin in SVM?
60. How do you interpret the coefficients in an SVM model?