General Linear Model:

1. What is the purpose of the General Linear Model (GLM)?

It describes the relationship between two variables and determining whether it is statistically significant; additionally, the model enables us to predict the value of the dependent variable given some new values of the independent variable(s).

1. What are the key assumptions of the General Linear Model?

It is having four core assumptions. They are

* Homoskedasticity – Same Variance
* Normally Distributed
* Linearly related
* Less or No multicollinearity

1. How do you interpret the coefficients in a GLM?

For every one-unit increase in [X variable], the [y variable] increases by [coefficient] when all other variables are held constant.

1. What is the difference between a univariate and multivariate GLM?

In multivariate tests the columns of Y are tested together, whereas in univariate tests the columns of Y are tested independently

1. Explain the concept of interaction effects in a GLM.

An interaction occurs when an independent variable has a different effect on the outcome depending on the values of another independent variable.

1. How do you handle categorical predictors in a GLM?

There are few methods to deal with categorical variables. They are:

* Label Encoding
* One hot encoding
* Combine Levels
* Dummy Encoding

1. What is the purpose of the design matrix in a GLM?

The purpose of the design matrix is to allow models that further constrain parameter sets.

1. How do you test the significance of predictors in a GLM?

We can use T-test for testing the significance of predictors in a GLM. If the p-value is less than 0.05 then a variable is said to be significant and have some relationship.

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

* Type I sum of squares are “sequential.”  In essence the factors are tested in the order they are listed in the model.
* Type III sum of squares are “partial.”  In essence, every term in the model is tested in light of every other term in the model.
* Type II sum of squares are similar to Type III, except that they preserve the principle of marginality.

1. Explain the concept of deviance in a GLM.

Deviance is a measure of error; lower deviance means better fit to data. The greater the deviance, the worse the model fits compared to the best case (saturated)

Regression:

1. What is regression analysis and what is its purpose?

Regression analysis is a statistical method that shows the relationship between two or more variables. Typically, a regression analysis is done for one of two purposes: In order to predict the value of the dependent variable for individuals for whom some information concerning the explanatory variables is available, or in order to estimate the effect of some explanatory variable on the dependent variable.

1. What is the difference between simple linear regression and multiple linear regression?

Simple Linear regression is having single predictor and with that we are predicting single target variable. Multiple Linear regression is having multiple predictors and with that we are predicting single or multiple target variables.

1. How do you interpret the R-squared value in regression?

an r-squared of 60% reveals that 60% of the variability observed in the target variable is explained by the regression model. Generally, a higher r-squared indicates more variability is explained by the model

1. What is the difference between correlation and regression?

The key difference between correlation and regression is that correlation measures the degree of a relationship between two independent variables (x and y). In contrast, regression is how one variable affects another.

1. What is the difference between the coefficients and the intercept in regression?

The simple linear regression model is essentially a linear equation of the form y = c + b\*x; where y is the dependent variable (outcome), x is the independent variable (predictor), b is the slope of the line; also known as regression coefficient and c is the intercept; labeled as constant.

1. How do you handle outliers in regression analysis?

There are few ways to handle the outliers. They are:

* Removing the outlier if removing it will not change distribution a lot.
* Cap the outlier with the Q1 -1.5 \* IQR or Q3 + 1.5 \* IQR
* Assign a Dummy Variable to Outliers

1. What is the difference between ridge regression and ordinary least squares regression?

when there is a difference in variance between predictor variables, OLS tends to give higher variance for coefficients corresponding to predictors with higher variance, while Ridge Regression reduces the variance differences between coefficients by shrinking them towards zero

1. What is heteroscedasticity in regression and how does it affect the model?

Heteroskedastic refers to a condition in which the variance of the residual term, or error term, in a regression model varies widely. Heteroscedasticity makes a regression model less dependable because the residuals should not follow any specific pattern

1. How do you handle multicollinearity in regression analysis?

* Remove some of the highly correlated independent variables.
* Linearly combine the independent variables, such as adding them together.
* Partial least squares regression uses principal component analysis to create a set of uncorrelated components to include in the model.
* LASSO and Ridge regression are advanced forms of regression analysis that can handle multicollinearity

1. What is polynomial regression and when is it used?

A polynomial regression model is a machine learning model that can capture non-linear relationships between variables by fitting a non-linear regression line, which may not be possible with simple linear regression. It is used when linear regression models may not adequately capture the complexity of the relationship

Loss function:

1. What is a loss function and what is its purpose in machine learning?

loss function is a measure of how good your prediction model does in terms of being able to predict the expected outcome(or value). We convert the learning problem into an optimization problem, define a loss function and then optimize the algorithm to minimize the loss function

1. What is the difference between a convex and non-convex loss function?

A convex function is one in which a line drawn between any two points on the graph lies on the graph or above it. There is only one requirement.

A non-convex function is one in which a line drawn between any two points on the graph may cross additional points. It was described as “wavy.”

When a cost function is non-convex, it has a higher chance of finding local minima rather than the global minimum, which is usually undesirable in machine learning models from an optimization standpoint.

1. What is mean squared error (MSE) and how is it calculated?

The Mean Squared Error measures how close a regression line is to a set of data points. It is a risk function corresponding to the expected value of the squared error loss.

To find the MSE, take the observed value, subtract the predicted value, and square that difference. Repeat that for all observations. Then, sum all of those squared values and divide by the number of observations. Notice that the numerator is the sum of the squared errors (SSE), which linear regression minimizes.

1. What is mean absolute error (MAE) and how is it calculated?

Mean Absolute Error (MAE) is calculated by taking the summation of the absolute difference between the actual and calculated values of each observation over the entire array and then dividing the sum obtained by the number of observations in the array.

1. What is log loss (cross-entropy loss) and how is it calculated?

Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label.

−(ylog(p)+(1−y)log(1−p))

1. How do you choose the appropriate loss function for a given problem?

There are various factors involved in choosing a loss function for specific problem such as type of machine learning algorithm chosen, ease of calculating the derivatives and to some degree the percentage of outliers in the data set.

1. Explain the concept of regularization in the context of loss functions.

Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting.

1. What is Huber loss and how does it handle outliers?

Huber regression is an example of a robust regression algorithm that assigns less weight to observations identified as outliers. To do so, it uses the Huber loss in the optimization routine. The Huber loss identifies outliers by considering the residuals, denoted by z. If the observation is considered to be regular, then apply the squared loss function. Otherwise, the observation is considered to be an outlier and you apply the absolute loss. Having said that, Huber loss is basically a combination of the squared and absolute loss functions.

1. What is quantile loss and when is it used?

As the name suggests, the quantile regression loss function is applied to predict quantiles. A quantile is the value below which a fraction of observations in a group falls. For example, a prediction for quantile 0.9 should over-predict 90% of the times.

1. What is the difference between squared loss and absolute loss?

Squared error is the difference between the predicted value and the actual value, squared. It is commonly used in regression analysis, as it has the desirable properties of being differentiable and non-negative. Squaring the difference also amplifies larger errors, which can be useful in certain situations. Absolute error, on the other hand, is the difference between the predicted value and the actual value, without squaring it. It is commonly used in measurement and instrumentation, as it is easy to interpret and understand. However, it does not have the same desirable properties as squared error, as it can be negative and is not differentiable. In summary, squared error is a measure of how far away the predicted value is from the actual value, but it is more sensitive to larger errors, while absolute error is a measure of how far away the predicted value is from the actual value, but it is more easy to interpret.

Optimizer (GD):

1. What is an optimizer and what is its purpose in machine learning?

 An optimizer is a function or an algorithm that adjusts the attributes of the neural network, such as weights and learning rates Thus, it helps in reducing the overall loss and improving accuracy.

1. What is Gradient Descent (GD) and how does it work?

Gradient descent is an optimization algorithm which is commonly-used to train machine learning models and neural networks. Training data helps these models learn over time, and the cost function within gradient descent specifically acts as a barometer, gauging its accuracy with each iteration of parameter updates.

1. What are the different variations of Gradient Descent?

There are three types of gradient descent learning algorithms: batch gradient descent, stochastic gradient descent and mini-batch gradient descent.

1. What is the learning rate in GD and how do you choose an appropriate value?

The learning rate gives you control of how big (or small) the updates are going to be. If you choose a learning rate that is too small, the gradient descent algorithm might take a really long time to find the minimum value of the error function. On the other hand, if you choose a learning rate that is too large, you might overshoot the minimum value of the error function, and may even never reach the optimal solution. The standard gradient descent procedure uses a fixed learning rate (e.g. 0.01) that is determined by trial and error.

1. How does GD handle local optima in optimization problems?

Steps should be made in proportion to the negative of the function gradient (move away from the gradient) at the current point to find local minima

In stochastic gradient descent the parameters are estimated for every observation, as opposed the whole sample in regular gradient descent (batch gradient descent). This is what gives it a lot of randomness. The path of stochastic gradient descent wanders over more places, and thus is more likely to "jump out" of a local minimum, and find a global minimum

1. What is Stochastic Gradient Descent (SGD) and how does it differ from GD?

As each iteration of the approach in GD requires computing the gradient of the cost function across the whole training dataset, GD takes some time to converge. SGD adjusts the model parameters more often than GD, which causes it to converge more quickly.

Due to the requirement to retain the whole training dataset, GD consumes a lot of memory.

As just one training sample needs to be stored for each iteration, SGD requires less memory.

1. Explain the concept of batch size in GD and its impact on training.

The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters. At the end of the batch, the predictions are compared to the expected output variables and an error is calculated. From this error, the update algorithm is used to improve the model.

1. What is the role of momentum in optimization algorithms?

Momentum is a strategy for accelerating the convergence of the optimization process by including a momentum element in the update rule. This momentum factor assists the optimizer in continuing to go in the same direction even if the gradient changes direction or becomes zero.

1. What is the difference between batch GD, mini-batch GD, and SGD?

|  |  |  |
| --- | --- | --- |
| **Gradient Descent** | **Stochastic Gradient Descent** | **Mini-batch gradient descent** |
| As each iteration of the approach requires computing the gradient of the cost function across the whole training dataset, GD takes some time to converge. | SGD adjusts the model parameters more often than GD, which causes it to converge more quickly | In order to strike a reasonable balance between speed and accuracy, the model parameters are changed more frequently than GD but less frequently than SGD. |
| Due to the requirement to retain the whole training dataset, GD consumes a lot of memory. | As just one training sample needs to be stored for each iteration, SGD requires less memory. | Just a percentage of the training samples had to be retained for each repetition, therefore the memory use is manageable. |
| GD is computationally expensive because the gradient of the cost function must be computed for the whole training dataset at each iteration. | As the cost function's gradient only needs to be calculated once for each repeat of training data, SGD is computationally efficient. | As the gradient of the cost function must be calculated for a portion of the training examples for each iteration, it is computationally efficient. |
| With little error, GD modifies the model's parameters based on the average of all training samples. | Due to the fact that SGD is updated using just one training sample, it has a lot of noise. | Mini-batch Gradient Descent has a significant amount of noise because the update is based on a small number of training examples. |

1. How does the learning rate affect the convergence of GD?

If the learning rate is too high, the algorithm may overshoot the minimum, and if it is too low, the algorithm may take too long to converge.

Regularization:

1. What is regularization and why is it used in machine learning?

Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting. Using Regularization, we can fit our machine learning model appropriately on a given test set and hence reduce the errors in it.

1. What is the difference between L1 and L2 regularization?

|  |  |
| --- | --- |
| **L1 Regularization** | **L2 Regularization** |
| The penalty term is based on the absolute values of the model's parameters. | The penalty term is based on the squares of the model's parameters. |
| Produces sparse solutions (some parameters are shrunk towards zero). | Produces non-sparse solutions (all parameters are used by the model). |
| Sensitive to outliers. | Robust to outliers. |
| Selects a subset of the most important features. | All features are used by the model. |
| Optimization is non-convex. | Optimization is convex. |
| The penalty term is less sensitive to correlated features. | The penalty term is more sensitive to correlated features. |
| Useful when dealing with high-dimensional data with many correlated features. | Useful when dealing with high-dimensional data with many correlated features and when the goal is to have a less complex model. |
| Also known as Lasso regularization. | Also known as Ridge regularization. |

1. Explain the concept of ridge regression and its role in regularization.

Ridge regression is a regression technique that adds a penalty term to the loss function to control the complexity of the model. It helps prevent overfitting by shrinking the coefficients towards zero.

1. What is the elastic net regularization and how does it combine L1 and L2 penalties?

Elastic net regularization combines L1 and L2 penalties to address the limitations of each. It balances between feature selection and coefficient shrinkage, allowing for selection of relevant features while dealing with correlated predictors.

1. How does regularization help prevent overfitting in machine learning models?

Regularization helps prevent overfitting by adding a penalty to the model's loss function, discouraging complex models that fit the training data too closely. It helps generalize the model to unseen data.

1. What is early stopping and how does it relate to regularization?

Early stopping is a technique where training is stopped before completion based on a validation set's performance. It helps prevent overfitting by finding the optimal point where the model has learned but not overfit the data.

1. Explain the concept of dropout regularization in neural networks.

Dropout regularization randomly sets a fraction of the input units to zero during training, forcing the network to learn more robust and generalized representations. It helps prevent overfitting by reducing co-adaptation between neurons.

1. How do you choose the regularization parameter in a model?

The regularization parameter is typically chosen using techniques like cross-validation or grid search. It involves evaluating the model's performance for different values of the parameter and selecting the one that yields the best trade-off between bias and variance.

1. What is the difference between feature selection and regularization?

Feature selection aims to select a subset of relevant features, while regularization controls the weights of all features. Feature selection directly removes irrelevant features, while regularization can shrink the coefficients of less important features towards zero.

1. What is the trade-off between bias and variance in regularized models?

The trade-off between bias and variance in regularized models is influenced by the regularization parameter. Increasing the regularization strength increases bias but decreases variance, leading to simpler models that generalize better but may have higher bias. Decreasing the regularization strength decreases bias but increases variance, allowing more complex models with potential overfitting.

SVM:

1. What is Support Vector Machines (SVM) and how does it work?

Support Vector Machines (SVM) is a machine learning algorithm used for classification and regression. It works by finding a hyperplane that separates data points into different classes, maximizing the margin between the classes.

1. How does the kernel trick work in SVM?

The kernel trick in SVM allows the algorithm to implicitly map data points to a higher-dimensional feature space. This enables SVM to find non-linear decision boundaries in the original input space.

1. What are support vectors in SVM and why are they important?

Support vectors are the data points that lie closest to the decision boundary in SVM. They play a crucial role in defining the decision boundary and determining the model's performance.

1. Explain the concept of the margin in SVM and its impact on model performance.

The margin in SVM is the region between the decision boundary and the nearest data points. A larger margin indicates better generalization and robustness of the model, reducing the risk of overfitting

1. How do you handle unbalanced datasets in SVM?

Unbalanced datasets in SVM can be handled by adjusting the class weights or using techniques like oversampling the minority class or under sampling the majority class to balance the training data.

1. What is the difference between linear SVM and non-linear SVM?

Linear SVM separates data using a linear decision boundary, while non-linear SVM uses kernel functions to map the data into a higher-dimensional space, allowing for non-linear decision boundaries.

1. What is the role of C-parameter in SVM and how does it affect the decision boundary?

The C-parameter in SVM controls the trade-off between achieving a larger margin and allowing misclassifications. Smaller values of C prioritize a larger margin, while larger values prioritize accurate classification of training points.

1. Explain the concept of slack variables in SVM.

Slack variables in SVM allow for misclassifications within the margin or on the wrong side of the decision boundary. They provide flexibility to handle overlapping or misclassified data points

1. What is the difference between hard margin and soft margin in SVM?

Hard margin SVM aims to find a decision boundary that perfectly separates the classes, assuming the data is linearly separable. Soft margin SVM allows for some misclassifications, introducing a penalty for misclassified points.

1. How do you interpret the coefficients in an SVM model?

The coefficients in an SVM model represent the importance or weight assigned to each feature. Positive coefficients indicate a positive contribution to the decision boundary, while negative coefficients indicate a negative contribution.

Decision Trees:

1. What is a decision tree and how does it work?

A decision tree is a flowchart-like model that makes decisions based on the values of input features. It works by splitting the data based on the features to create branches of decisions.

1. How do you make splits in a decision tree?

Splits in a decision tree are made by selecting the feature and the threshold value that best separates the data into different classes or groups based on certain criteria.

1. What are impurity measures (e.g., Gini index, entropy) and how are they used in decision trees?

Impurity measures like Gini index and entropy are used to quantify the homogeneity of a node in a decision tree. They help in determining the optimal splits by measuring the uncertainty or disorder in the data.

1. Explain the concept of information gain in decision trees.

Information gain measures the reduction in entropy or impurity achieved by splitting a node based on a particular feature. It helps in selecting the feature that provides the most useful information for classification or prediction.

1. How do you handle missing values in decision trees?

Missing values in decision trees can be handled by using techniques like surrogate splits, which consider alternative features when a missing value is encountered, or by assigning the missing values to the most probable class or value.

1. What is pruning in decision trees and why is it important?

Pruning in decision trees is the process of removing unnecessary branches or nodes to improve the model's generalization and avoid overfitting. It helps in simplifying the tree and reducing complexity.

1. What is the difference between a classification tree and a regression tree?

A classification tree is used for categorical or discrete target variables, while a regression tree is used for continuous target variables. Classification trees predict classes, while regression trees predict numerical values.

1. How do you interpret the decision boundaries in a decision tree?

Decision boundaries in a decision tree are the points where the tree makes a decision to classify or predict. They are represented by the splits or branches in the tree that separate the data based on feature values.

1. What is the role of feature importance in decision trees?

Feature importance in decision trees measures the relative significance of each feature in making accurate predictions. It helps in understanding the influence of different features on the target variable.

1. What are ensemble techniques and how are they related to decision trees?

Ensemble techniques combine multiple decision trees to make more accurate predictions. They use methods like bagging, boosting, and random forests to improve the overall performance and robustness of the model.

Ensemble Techniques:

1. What are ensemble techniques in machine learning?

Ensemble techniques in machine learning combine multiple models to make predictions. They can be used to improve accuracy, reduce overfitting, and handle complex problems.

1. What is bagging and how is it used in ensemble learning?

Bagging is an ensemble technique where multiple models are trained independently on different subsets of the training data. The predictions from each model are then combined to make the final prediction.

1. Explain the concept of bootstrapping in bagging.

Bootstrapping in bagging is the process of creating multiple subsets of the training data by randomly sampling with replacement. It helps in generating diverse subsets for training different models in bagging.

1. What is boosting and how does it work?

Boosting is an ensemble technique that trains models sequentially, where each model learns from the mistakes of the previous model. It assigns weights to the training samples to focus on the difficult-to-classify instances.

1. What is the difference between AdaBoost and Gradient Boosting?

AdaBoost and Gradient Boosting are boosting algorithms. AdaBoost adjusts the weights of misclassified samples, while Gradient Boosting builds models to minimize the loss function by adding models iteratively.

1. What is the purpose of random forests in ensemble learning?

Random forests are an ensemble technique that combines multiple decision trees. They introduce randomness in feature selection and bagging to create diverse trees and improve prediction accuracy.

1. How do random forests handle feature importance?

Random forests calculate feature importance by measuring the average decrease in impurity or entropy caused by each feature across all the trees. Features with higher importance contribute more to the predictions.

1. What is stacking in ensemble learning and how does it work?

Stacking in ensemble learning combines predictions from multiple models as input to a meta-model. It trains a higher-level model to make predictions based on the outputs of the individual models.

1. What are the advantages and disadvantages of ensemble techniques?

The advantages of ensemble techniques include improved accuracy, reduced overfitting, handling complex problems, and robustness to noisy data. Disadvantages include increased complexity and computational resources required.

1. How do you choose the optimal number of models in an ensemble?

The optimal number of models in an ensemble depends on the specific problem and dataset. It can be determined using techniques like cross-validation, monitoring performance metrics, and considering computational resources.