Answer Key.

1. **R-squared or Residual Sum of Squares (RSS) which one of these two is a better measure of the goodness of fit model in regression and why?**

Ans. In general, the regression model is believed to fit the data better when measured using R-squared or coefficient of determination than residual square (RSS). R-squared indicates the percentage of variance in variance explained by individual variables. Although RSS only calculates the total squared variance of the observed and predicted values, ignoring the percentage of explained variance, a higher R-squared value indicates a better match. R-squared provides a more comprehensive assessment of how well the model fits the data.

2. **What are TSS (Total Sum of Squares), ESS (Explained Sum of Squares), and RSS (Residual Sum of Squares) in regression? Also, mention the equation relating these three metrics with each other.**

Ans. Population change in the dependent variable (Y) is represented by the population sum of squares (TSS) in the regression analysis. It measures how much the value of the variable differs from the mean. The value of the variable as explained by the regression model is measured by the ESS (or explained sum of squares). It measures the proportion of the total variation in Y that can be explained by the independent variable (X). The amount of unexplained variation in the variance of the regression model is captured by the RSS (residual sum of squares). Calculate the sum of squares of the differences between the observed values of Y and the values predicted by the regression equation.

TSS = ESS + RSS is the combined equation of these three measurements.

The difference between the variance explained by the regression model (ESS) and the residual unexplained variance (RSS) are two definitions of labeling all changes that occur in the equation. TSS is the sum of these two points and provides a comprehensive view of the total variance of the variable and its distribution between explained and unexplained variables.

3. **What is the need for regularization in machine learning?**

Ans. Regularization is an important part of machine learning to avoid overfitting, which occurs when the model learns to capture noise or random oscillations in the training data instead of the base model. Regularization is necessary because the model is very flexible but very difficult to match training very well, making it bad for new equipment. Regularization methods such as L1 (Lasso) and L2 (Ridge) constrain the model during training, preventing the development of models that are too complex and reducing the likelihood of overfitting. Regularization helps the overall model for new unknown data, improving its performance and simplicity by penalizing large values or simplifying the model.

By reducing the coefficients of less significant factors, the moderation function can also reduce the possibility of multicollinearity, which occurs when predictor variables are different. Regularization methods also help select specific features from the analysis and show the most important of the best estimates by pushing some coefficients to zero. In addition to making the model easier to understand, this also reduces computational and memory requirements. In summary, continuous operation is a good way to balance the complexity structure and the overall operation; this can increase the reliability, interpretability, and effectiveness of machine learning models in real-world environments.

4. **What is the Gini–impurity index?**

Answer Key.

Ans. In decision tree algorithms, the Gini Impurity Index is a metric used to measure the inconsistency or impurity of data recorded specifically in the classification function. It measures the probability that a randomly selected object will be misclassified if its label is randomly assigned based on the distribution of letters in the set.

In mathematics, add the square probability of each selected category. Gini is subtracted to determine impurity. A collection is purer if it has a lower Gini impurity; This means it has the most instances of a category.

The decision tree reduces the Gini impurity of the result by splitting the nodes according to their properties to create different types of child nodes. The process is performed iteratively until stopping is achieved or further reduction of impurities is not possible.

The Gini Impurity Index varies between 0 and 1; where 1 represents the highest impurity (equal distribution of products in all categories) and 0 represents perfect purity (all elements are in the same category). It is a widely used measure of impurity because it can easily and effectively guide the construction of decision trees. Decision trees can efficiently partition space to identify distinct clusters by selecting partitions that minimize Gini impurities. This process ultimately produces accurate classification results. The efficiency of the Gini Impurity Index calculation also makes it suitable for ad-hoc machine learning and big data.

5. Are unregularized decision trees prone to overfitting? If yes, why?

Ans. Regular decision trees have compatibility problems. When the model learns to detect noise or random oscillations in the training data instead of the base model, it is said to perform well and perform poorly when applied to new data.

The modification is very good and capable of capturing the interaction between features. Therefore, they can become complicated by creating partitions and branches that fit the training data, which may contain noise and outliers. Due to its complexity, the model can remember the training set but not the base model.

Regular decision trees can be too deep, too specialized on the training set, and perform poorly on new, untested data. Regularization techniques such as tree depth limiting or pruning are important to prevent overfitting as they control model complexity and encourage the development of more direct and widely used models.

6. What is an ensemble technique in machine learning?

Ans. Hybrid techniques in machine learning combine multiple models to increase the accuracy of predictions. Combined methods use combinations of multiple models rather than relying on a single model to provide more accurate predictions.

Combination methods include stacking, random forest, boosting, and bagging. Bagging trains multiple instances of the same learning algorithm using various subsets of the training data and averages its predictions. Weaker students are trained backward, with improvements focusing on situations not classified in previous models. The decision tree model and bagging are combined in a random forest where specific features and data are used to train multiple trees. Stacking trains a meta-model on top of multiple models, integrating the predictions of these models.

Answer Key.

The integrated system can improve overall performance, increase stability, and reduce workload. They are frequently used in a variety of applications because they can handle complex data structures and provide high-accuracy predictions.

7. **What is the difference between Bagging and Boosting techniques?**

Ans. Although two aggregation methods (Bagging, Bootstrap Aggregating, and Boosting) are used to improve prediction in machine learning, they use different methods to combine multiple models.

Bagging (Bootstrap Aggregating):

a. Using various subsets of the training data, bagging needs to be trained multiple times on the same learning algorithm, sometimes with modifications.
b. Each model is trained independently and the results are often combined through voting or averaging.
c. Bag creates a variety of models, each capturing characteristics of the data; this helps reduce overfitting and variation.
d. An example is Random Forest, which combines decision trees with bagging.

Boosting:

a. The boosting algorithm targets initial cases where the model is misclassified and retrains weak learners (models that perform better than chance).
b. All weak students are taught to correct mistakes made by their superiors by referring to incorrect situations.
c. Boosting improves the model to reduce bias and increase the accuracy of predictions.
d. AdaBoost, Gradient Boosting Machine (GBM), and XGBoost are a few examples.
e. In summary, the main difference between the two is the way they train the patterns: Bagging creates multiple patterns on its own while boosting reproduces the pattern of patterns by correcting flaws in the initial patterns.

8. What is an out-of-bag error in random forests?

Ans. Out-of-bag (OOB) error in random forests is calculated using self-reported data to estimate how well the model will handle missing data. For each decision tree in a random forest, some of the training data is selected and modified. This indicates that the training subset of a tree does not contain all data points.

Out-of-bag error is calculated by analyzing each sample of data points using trees that exclude only the data points of the training subset (benchmark, i.e. contents outside the bag). The prediction error is then calculated by comparing the canned predictions with the actual text of the data points.

Without separate validation, the out-of-bag error can provide an unbiased measure of the standard error because each data point represents an out-of-bag sample of approximately one-third of the wood (on average). It is a useful tool for evaluating the performance of random forests and tuning hyperparameters when training a model.

9. What is K-fold cross-validation?

Ans. K-fold cross-validation is an example used to test the performance of machine learning models. Split the dataset into K multiples or subsets of equal parts and train and test K times.

Answer Key.

One of the K folds becomes validation for each iteration and the remaining K-1 folds are used for training. During K iterations of this function, each fold runs exactly once as a set guarantee. Performance metrics (such as accuracy or error) are averaged over K iterations to obtain an overall estimate of model performance.

10. What is hyperparameter tuning in machine learning and why is it done?
Ans. In machine learning, selecting the best hyperparameters for a particular model is called hyperparameter tuning. Hyperparameters are predetermined settings rather than learned from data during training. The number of layers in the neural network, the depth of the decision tree, and learning gradient descent are some examples.

Hyperparameter tuning is important because the choice of hyperparameters can affect the performance and feasibility of the model. Inappropriate selection of hyperparameter values will yield less than optimal results, such as overfitting or underfitting. Therefore, hyperparameter tuning aims to define a set of hyperparameters that will maximize the model's performance on hypothetical data.

Methods such as grid search, random search, Bayesian optimization, and genetic algorithms are all examples of hyperparameter tuning techniques. This technique explores the hyperparameter space by analyzing various combinations of hyperparameters and selecting the combination that produces the best value as the selected parameter.

Hyperparameter tuning for the model is an important factor in improving the overall performance, stability, and reliability of the model by correcting errors that control the behavior and learning of machine learning.

11. What issues can occur if we have a large learning rate in Gradient Descent?
Ans. If we have a large learning rate in gradient descent, several issues can arise:

a. Overshoot: When a larger training value is used, the method will exceed the minimum value of the function. During training, the algorithm may change or diverge rather than converge to the minimum, resulting in poor and unstable behavior.

b. Unstable: The update of the model is not stable due to the high learning rate. Because of this instability, convergence to the optimal solution can be difficult, especially in areas where the slope changes rapidly.

c. Difference: This method will not be possible for all subjects. The cost function can continuously grow as the optimization model grows and gives rise to different optimization methods.

12. Can we use Logistic Regression for the classification of Non-Linear Data? If not, why?
Ans. Although logistic regression is essentially a linear distribution, it can also be used to distribute nonlinear data. However, it is still possible to capture some relationships between features and target variables using methods such as polynomial feature expansion or basis function expansion, or by building construction.

Answer Key.

For example, if the connection between the features and the target matches in a different non-linear way, we can create additional features using polynomials or interaction terms of the original features. This makes it possible to model biased decision-making using logistic regression.

Additionally, techniques such as using kernel methods or nonlinear functions in neural networks can also enable logistic regression to handle nonlinear data by changing the location to high places where the data are linearly separated.

Although simple nonlinearities can be handled using logistic regression, it is important to note that more complex interactions such as decision trees, support vector machines, or neural network methods may require more models.

13. Differentiate between Adaboost and Gradient Boosting.
Ans. Adaboost (Adaptive Boosting) and Gradient Boosting are both ensemble learning methods used to improve the performance of weak learners (models that perform slightly better than random chance), but they differ in their approach to combining multiple models.

Adaboost (Adaptive Boosting):

a. By creating weak students one after another, Adaboost increases interest in situations that isolate early learners.
b. Adaboost gives more weight to misclassified examples with each iteration to ensure that the next model prioritizes correctly classifying these examples.
c. The final prediction is made by summing the predictions of each weak learner using the weight, with more weight given to the more accurate model.
d. Adaboost is particularly successful when used for weak learners such as decision stumps (single-pane decision trees) because it is less competitive than the other method.

Gradient Boosting:

a. By gradually creating weaker learners, gradient boosting minimizes the performance loss of each learner relative to the rest of the previous model.
b. Compared to Adaboost, gradient boosting optimizes model parameters (such as tree models and leaf values), fitting each student to all residuals to eliminate errors greedily.
c. Gradient boosting typically uses gradient descent optimization techniques to change sample parameters to reduce the overall loss of the cluster.
d. Some well-known implementations of gradient boosting include LightGBM, XGBoost, and Gradient Boosting Machines (GBM).

14. What is the bias-variance trade-off in machine learning?
Ans. The bias-variance tradeoff is a fundamental concept in machine learning that describes the relationship between a model's bias, variance, and overall predictive performance.

Bias: The error that occurs when a simple model is used to predict the actual situation is called bias. When a high-bias model underfits the data, it oversimplifies the underlying model and misses the true connection between target variables and traits. Large instances of bias often produce large learning errors, and the complexity of the data is very easy to capture.

Answer Key.

Variation: This describes how sensitive the model is to small changes or noise in the training process. The high variance model tends to misrepresent the data; This is bad for unrecognized data as it captures noise and random oscillations in the training process. A significant change has a significant bug but involves minimal training.

15. Give a short description of each Linear, RBF, and Polynomial kernel used in SVM.
Ans.

Linear Kernel:

a. The simplest kernel used in SVM is the linear kernel.
b. Calculates the product of feature vectors in the input field.
c. Best for linearly separable data or where there are many specifications regarding the number of samples.
d. It works well when there is a relationship between different targets and features.

RBF (Radial Basis Kernel) Kernel:

a. The RBF kernel is nonlinear in SVM.
b. The method involves presenting input data to a high-dimensional space and calculating the distance or similarity between feature vectors.
c. Its special feature is to capture complex non-linear correlations between features and differences between target variables.
d. It has two hyperparameters: C (corrects the trade-off between edges and model complexity) and gamma (determines the width of the Gaussian function).

Polynomial Kernel:

a. The product of the eigenvectors raised to the given power (represented by the index) is calculated by the polynomial table.
b. Allows SVM to capture nonlinear relationships by changing the input space to a higher space.
c. Works well for data without boundary conditions, but can be affected by overfitting and sensitivity to hyperparameter selection.