Machine Learning

Supervised ML:

# Regression

## Linear Regression

Notes:

Linear regression is a type of supervised machine learning algorithm used for predicting continuous numerical values. It is a statistical approach that models the relationship between a dependent variable (output) and one or more independent variables (inputs) by fitting a linear equation to the observed data.

Here are some key points about linear regression:

* Linear regression assumes a linear relationship between the independent variables and the dependent variable.
* The goal of linear regression is to find the best-fitting line (or hyperplane in higher dimensions) that minimizes the **sum of the squared difference**s between the predicted and actual values.
* The line is defined by the coefficients (slope and intercept) that determine the relationship between the independent variables and the dependent variable.
* Linear regression can be used for both simple regression (with one independent variable) and multiple regression (with multiple independent variables).
* It is important to assess the quality of the linear regression model using evaluation metrics such as mean squared error (MSE), root mean squared error (RMSE), and coefficient of determination (R-squared).

In summary, linear regression is a widely used algorithm for predicting continuous values based on the linear relationship between input variables and the output variable.

**Coefficients:**

**Y=MX+C**

Suppose we have a dataset of housing prices, where the independent variable (input) is the size of the house (in square feet) and the dependent variable (output) is the price of the house (in dollars). We want to build a linear regression model to predict house prices based on their sizes.

After training the linear regression model on the dataset, we obtain the following equation:

Price = 50,000 + 100 \* Size

In this equation, the coefficient of the Size variable is 100. This means that for every one-unit increase in the size of the house, the price is expected to increase by $100. The coefficient of 50,000 represents the intercept or the base price of a house with a size of 0 square feet.

So, in this example, the coefficients tell us the relationship between the size of the house and its price, allowing us to make predictions on new houses based on their sizes.

**Error metrics in regression:**

Sure! In linear regression, there are several error metrics that are commonly used to evaluate the performance of the model. These metrics help us understand how well the model is fitting the data and how accurate its predictions are. Let's dive into each of these error metrics in detail:

1. Sum of Squared Error (SSE): SSE measures the sum of the squared differences between the actual target values and the predicted values. It quantifies the overall error of the model. A lower SSE indicates a better fit of the model to the data.
2. Total Sum of Squares (TSS): TSS represents the total variation in the target variable. It measures the sum of the squared differences between the actual target values and the mean of the target variable. TSS provides a baseline to compare the performance of the model. A lower TSS indicates a better fit of the model.
3. Coefficient of Determination (R2): R2 is a statistical measure that represents the proportion of the variance in the target variable that can be explained by the independent variables in the model. It ranges from 0 to 1, where 0 indicates that the model does not explain any of the variance and 1 indicates a perfect fit. A higher R2 value indicates a better fit of the model.
4. Mean Squared Error (MSE): MSE is another commonly used error metric in linear regression. It calculates the average of the squared differences between the actual target values and the predicted values. MSE provides a measure of the average squared deviation of the predictions from the actual values. A lower MSE indicates a better fit of the model.
5. Root Mean Squared Error (RMSE): RMSE is the square root of the MSE. It provides a measure of the average deviation of the predictions from the actual values, in the same units as the target variable. RMSE is often preferred over MSE as it is more interpretable and easier to relate to the original scale of the target variable. Like MSE, a lower RMSE indicates a better fit of the model.
6. Mean Absolute Error (MAE) is another commonly used error metric in regression tasks, including linear regression. It measures the average absolute difference between the actual target values and the predicted values. MAE provides a measure of the average deviation of the predictions from the actual values, regardless of the direction (positive or negative). A lower MAE indicates a better fit of the model.

Here are some key points about MAE:

MAE is calculated by taking the average of the absolute differences between the predicted and actual values.It is less sensitive to outliers compared to MSE and RMSE because it does not involve squaring the differences.MAE is expressed in the same units as the target variable, making it more interpretable and easier to relate to the original scale.Like MSE and RMSE, the goal is to minimize MAE to improve the accuracy of the linear regression model.

These error metrics help us assess the performance of the linear regression model and compare different models. By analyzing these metrics, we can determine how well the model is capturing the relationship between the independent variables and the target variable.

It's important to note that these error metrics are not the only ones used in linear regression. Depending on the specific problem and requirements, other metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) can also be used.

**Residuals:**

Residuals, in the context of linear regression, refer to the differences between the actual values of the target variable and the predicted values by the regression model. They represent the unexplained variation or error in the model's predictions.

Here are some key points about residuals:

* Residuals are calculated by subtracting the predicted values from the actual values.
* They can be positive or negative, depending on whether the predicted value is higher or lower than the actual value.
* Residuals provide insights into how well the regression model fits the data. Ideally, the residuals should be as small as possible, indicating a good fit.
* By analyzing the residuals, you can identify patterns or trends that the model might have missed, such as non-linear relationships or heteroscedasticity (unequal variance).
* Residual analysis is an important step in evaluating the assumptions of linear regression, such as normality, independence, and constant variance.
* Plotting the residuals against the predicted values or the independent variables can help identify any systematic patterns or outliers in the data.

In summary, residuals represent the discrepancies between the actual and predicted values in a linear regression model. Analyzing residuals can provide insights into the model's performance and help identify areas for improvement.

**Assumptions of Linear Regression:**

The assumptions of linear regression are important to ensure the validity and reliability of the model's results. Here are the key assumptions:

1. Linearity: The relationship between the independent variables and the dependent variable is linear. This means that the effect of each independent variable on the dependent variable is constant across all levels of the independent variables.
2. Independence: The observations in the dataset are independent of each other. There should be no correlation or dependence between the residuals or errors of the model.
3. Homoscedasticity: The variance of the residuals is constant across all levels of the independent variables. In other words, the spread of the residuals should be the same for all predicted values.\*Homoscedasticity\*

The assumption of \*homoscedasticity\* (constant variance), is crucial to linear regression models. \*Homoscedasticity\* describes a situation in which the error term or variance or the "noise" or random disturbance in the relationship between the independent variables and the dependent variable is the same across all values of the independent variable. In other words, there is a constant variance present in the response variable as the predictor variable increases. If the "n oise" is not the same across the values of an independent variable, we call it \***heteroscedasticity**\*, opposite of \***homoscedasticity**\*.

1. Normality: The residuals follow a normal distribution. This assumption is important for hypothesis testing, confidence intervals, and prediction intervals.
2. No multicollinearity: The independent variables are not highly correlated with each other. High multicollinearity can lead to unstable and unreliable estimates of the regression coefficients.
3. No endogeneity: There is no relationship between the error term and the independent variables. Endogeneity can occur when there is reverse causality or omitted variable bias.

It is important to check these assumptions before interpreting the results of a linear regression model. Violations of these assumptions can lead to biased and inefficient estimates, as well as incorrect inferences.

**Modeling best practices for LinearRegression:**

* Using a cost function to fit the linear regression model is indeed a modeling best practice. It helps in finding the optimal values for the regression coefficients by minimizing the difference between the predicted values and the actual values of the dependent variable.
* Developing multiple models is also a good practice in linear regression. By trying out different combinations of independent variables, transformations, or interactions, you can explore various potential relationships and identify the most suitable model for your data.
* After developing multiple models, it is important to compare the results to determine which model fits your data and your specific goals. This comparison can be based on various evaluation metrics such as the coefficient of determination (R-squared), mean squared error (MSE), or other relevant measures. The choice of the best model depends on whether you prioritize prediction accuracy or interpretability.

By following these best practices, you can improve the accuracy and reliability of your linear regression models.

### **Regularization Techniques**

* Three sources of error for your model are: bias, variance, and irreducible error.
* Regularization is a way to achieve building simple models with relatively low error. It helps you avoid overfitting by penalizing high-valued coefficients. It reduces parameters and shrinks the model.
* Regularization adds an adjustable regularization strength parameter directly into the cost function.
* Regularization performs feature selection by shrinking the contribution of features, which can prevent overfitting.
* In Ridge Regression, the complexity penalty λ is applied proportionally to squared coefficient values.
  + The penalty term has the effect of “shrinking” coefficients toward 0.
  + This imposes bias on the model, but also reduces variance.
  + We can select the best regularization strength lambda via cross-validation.
  + It’s a best practice to scale features (i.e. using StandardScaler) so penalties aren’t impacted by variable scale.
* In LASSO regression: the complexity penalty λ (lambda) is proportional to the absolute value of coefficients. LASSO stands for : Least Absolute Shrinkage and Selection Operator.
  + Similar effect to Ridge in terms of complexity tradeoff: increasing lambda raises bias but lowers variance.
  + LASSO is more likely than Ridge to perform feature selection, in that for a fixed λ, LASSO is more likely to result in coefficients being set to zero.
* Elastic Net combines penalties from both Ridge and LASSO regression. It requires tuning of an additional parameter that determines emphasis  of L1 vs. L2 regularization penalties.
* LASSO’s feature selection property yields an interpretability advantage, but may underperform if the target truly depends on many of the features.
* Elastic Net, an alternative hybrid approach, introduces a new parameter α (alpha) that determines a weighted average of L1 and L2 penalties.
* Regularization techniques have an analytical, a geometric, and a probabilistic interpretation.

**Views of Regularization:**

Regularization is a technique used in machine learning to prevent overfitting, which is when a model becomes too complex and starts to perform poorly on new data. It helps us find a balance between having a model that fits the training data well and a model that generalizes well to new data.

There are different types of regularization techniques, such as Ridge and LASSO. These techniques work by adding a penalty term to the model's cost function, which encourages the model to have smaller coefficients. Smaller coefficients mean that the model is simpler and less likely to overfit.

To understand regularization intuitively, we can think of it in a few different ways:

1. Analytic view: Regularization forces the coefficients of the model to be smaller, which restricts their plausible range. This leads to a simpler model with lower variance.
2. Geometric view: Regularization can be visualized as finding the intersection between the penalty boundary and the contour of the cost function. This helps us choose the coefficients that minimize both the error and the regularization term. Ridge tends to have a circular penalty boundary, while LASSO tends to have a diamond-shaped boundary. This geometric difference is why LASSO tends to zero out certain coefficients.
3. Probabilistic view: Regularization can be seen as a way to recalibrate our understanding of the coefficients. It introduces prior distributions for the coefficients, which helps us choose the coefficients that minimize both the error and the regularization term.

**RidgeL2 and LassoL1 to perform Regularization:**

Ridge and LASSO regularization techniques have different penalty boundaries and impacts on coefficients. Here's a summary of their differences:

Ridge regularization:

* Penalty boundary: The penalty boundary of Ridge regularization is circular in shape.
* Impact on coefficients: Ridge regularization shrinks the coefficients towards zero but does not set them exactly to zero. It reduces the magnitude of coefficients without eliminating any of them completely. The coefficients can take any value within the circular boundary.

LASSO regularization:

* Penalty boundary: The penalty boundary of LASSO regularization is diamond-shaped.
* Impact on coefficients: LASSO regularization not only shrinks the coefficients but also performs feature selection by setting some coefficients to exactly zero. It effectively eliminates certain coefficients, leading to a sparse model. The coefficients can take any value within the diamond-shaped boundary, but some coefficients will be exactly zero.

In summary, Ridge regularization allows all coefficients to shrink but not become zero, while LASSO regularization can set some coefficients to exactly zero, effectively performing feature selection. The choice between Ridge and LASSO regularization depends on the specific problem and the desired outcome.

Explanation for the practice questions:

1. **What is linear regression?**

Linear regression is a supervised machine learning algorithm used to predict a continuous target variable based on one or more input features.

1. **What are the assumptions of linear regression?**

The assumptions of linear regression include linearity, independence, homoscedasticity, and normality of residuals.

1. **How do you interpret the coefficients in linear regression?**

The coefficients in linear regression represent the change in the target variable for a one-unit change in the corresponding input feature, assuming all other variables are held constant.

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

Simple linear regression involves predicting a target variable using only one input feature, while multiple linear regression involves predicting the target variable using multiple input features.

1. **How do you evaluate the performance of a linear regression model?**

The performance of a linear regression model can be evaluated using metrics such as mean squared error (MSE), root mean squared error (RMSE), and R-squared.

1. **What is the purpose of the cost function in linear regression?**

The cost function in linear regression measures the difference between the predicted values and the actual values, and it is used to optimize the model parameters during the training process.

1. **How can you handle categorical variables in linear regression?**

Categorical variables can be handled in linear regression by using techniques such as one-hot encoding or ordinal encoding to convert them into numerical values.

1. **What is the difference between correlation and regression?**

Correlation measures the strength and direction of the linear relationship between two variables, while regression predicts the value of a dependent variable based on the values of independent variables.

1. **What is the purpose of regression analysis in machine learning?**
   * Regression analysis is used to model the relationship between a dependent variable and one or more independent variables. It helps in predicting continuous numerical values and understanding the impact of independent variables on the dependent variable.
2. **Explain the concept of linear regression and its assumptions.**
   * Linear regression is a statistical approach to modeling the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship between the variables and that the errors are normally distributed and have constant variance.
3. **What is the difference between simple linear regression and multiple linear regression?**
   * Simple linear regression involves only one independent variable, while multiple linear regression involves two or more independent variables. Simple linear regression aims to find a linear relationship between the dependent variable and a single independent variable, whereas multiple linear regression considers the combined effect of multiple independent variables on the dependent variable.
4. **How do you interpret the coefficients in a linear regression model?**
   * The coefficients in a linear regression model represent the change in the dependent variable for a one-unit change in the corresponding independent variable, holding all other variables constant. A positive coefficient indicates a positive relationship, while a negative coefficient indicates a negative relationship.
5. **What is the purpose of data splitting in regression analysis?**
   * Data splitting is done to evaluate the performance of a regression model. The dataset is divided into training and testing sets. The training set is used to train the model, while the testing set is used to assess how well the model generalizes to unseen data.
6. **Describe the process of cross-validation and its advantages.**
   * Cross-validation is a technique used to assess the performance of a model by splitting the data into multiple subsets. It involves training the model on a subset of the data and evaluating it on the remaining subset. This process is repeated multiple times, and the average performance is calculated. Cross-validation helps in estimating the model's performance on unseen data and reduces the risk of overfitting.
7. **What is the bias-variance trade-off in machine learning?**
   * The bias-variance trade-off refers to the trade-off between the model's ability to fit the training data (low bias) and its ability to generalize to unseen data (low variance). A model with high bias may underfit the data, while a model with high variance may overfit the data. Finding the right balance is crucial for building a good predictive model.
8. **Explain the concept of regularization and its role in regression models.**
   * Regularization is a technique used to prevent overfitting in regression models. It adds a penalty term to the loss function, which discourages the model from fitting the training data too closely. Regularization helps in reducing the model's complexity and improving its generalization performance.
9. **What are the differences between Ridge, LASSO, and Elastic Net regularization techniques?**
   * Ridge regularization adds the sum of squared coefficients multiplied by a regularization parameter to the loss function. It shrinks the coefficients towards zero but does not set them exactly to zero.
   * LASSO regularization adds the sum of absolute values of coefficients multiplied by a regularization parameter to the loss function. It can set some coefficients exactly to zero, effectively performing feature selection.
   * Elastic Net regularization is a combination of Ridge and LASSO regularization. It adds both the sum of squared coefficients and the sum of absolute values of coefficients multiplied by regularization parameters to the loss function. It provides a balance between Ridge and LASSO regularization.
10. **How can you evaluate the performance of a regression model using R-squared?**
    * R-squared, also known as the coefficient of determination, is a metric used to evaluate the goodness of fit of a regression model. It measures the proportion of the variance in the dependent variable that is predictable from the independent variables. R-squared ranges from 0 to 1, where 1 indicates a perfect fit. Higher R-squared values indicate better model performance.

## **Classification:**

## **Classification Problems**

The two main types of supervised learning models are:

* Regression models, which predict a continuous outcome
* Classification models, which predict a categorical outcome.

The most common models used in supervised learning are:

* Logistic Regression
* K-Nearest Neighbors
* Support Vector Machines
* Decision Tree
* Neural Networks
* Random Forests
* Boosting
* Ensemble Models

With the exception of logistic regression, these models are commonly used for both regression and classification. Logistic regression is most common for dichotomous and nominal dependent variables.

## **Logistic Regression:**

Logistic regression is a type of regression that models the probability of a certain class occurring given other independent variables. It uses a logistic or logit function to model a dependent variable. It is a very common predictive model because of its high interpretability.

## **Classification Error Metrics**

A confusion matrix tabulates true positives, false negatives, false positives and true negatives. Remember that the false positive rate is also known as a type I error. The false negatives are also known as a type II error.

Accuracy is defined as the ratio of true postives and true negatives divided by the total number of observations. It is a measure related to predicting correctly positive and negative instances.

Recall or sensitivity identifies the ratio of true positives divided by the total number of actual positives. It quantifies the percentage of positive instances correctly identified.

Precision is the ratio of true positive divided by total of predicted positives. The closer this value is to 1.0, the better job this model does at identifying only positive instances.

Specificity is the ratio of true negatives divided by the total number of actual negatives. The closer this value is to 1.0, the better job this model does at avoiding false alarms.

The receiver operating characteristic (ROC) plots the true positive rate (sensitivity) of a model vs. its false positive rate (1-sensitivity).

The area under the curve of a ROC plot is a very common method of selecting a classification methods.

The precision-recall curve measures the trade-off between precision and recall.

The ROC curve generally works better for data with balanced classes, while the precision-recall curve generally works better for data with unbalanced classes.

### **A diagram of a function Description automatically generated**

### **KNN(K-Nearest Neighbor):**

## **K Nearest Neighbor Methods for Classification**

K nearest neighbor methods are useful for classification. The elbow method is frequently used to identify a model with low K and low error rate.

These methods are popular due to their easy computation and interpretability, although it might take time scoring new observations, it lacks estimators, and might not be suited for large data sets.

A screen shot of a diagram

Description automatically generated

### **SVM(support vector machine):**

The main idea behind support vector machines is to find a hyperplane that separates classes by determining decision boundaries that maximize the distance between classes.

When comparing logistic regression and SVMs, one of the main differences is that the cost function for logistic regression has a cost function that decreases to zero, but rarely reaches zero. SVMs use the Hinge Loss function as a cost function to penalize misclassification. This tends to lead to better accuracy at the cost of having less sensitivity on the predicted probabilities.

Regularization can help SVMs generalize better with future data.

By using gaussian kernels, you transform your data space vectors into a different coordinate system, and may have better chances of finding a hyperplane that classifies well your data.SVMs with RBFs Kernels are slow to train with data sets that are large or have many features.

### **Decision Trees:**

Decision trees split your data using impurity measures. They are a greedy algorithm and are not based on statistical assumptions.

The most common splitting impurity measures are Entropy and Gini index.Decision trees tend to overfit and to be very sensitive to different data.

Cross validation and pruning sometimes help with some of this.

Great advantages of decision trees are that they are really easy to interpret and require no data preprocessing.

Decision tree model outputs a set of rules, and each rule is a If-else chain from root to a leaf node. Decision tree mimics human reasoning process which makes it very intuitive to human users and makes it a high-interpretable model.

Entropy and Gini index are 2 criteria or error metrics in decision trees

Max\_depth and min\_samples are 2 more hyperparameters passed to DecisionTreeClassifier

To determine the split for each node of a decision tree, you typically use a metric to measure the "goodness" of a split. Here are the steps:

1. **Calculate the impurity of the parent node**: This can be done using different metrics. Two common ones are Gini Impurity and Entropy. Both of these metrics give a measure of how "mixed" the classes in the parent node are.
2. **For each possible split, calculate the impurity of the child nodes**: Again, this can be done using the same metric (Gini Impurity or Entropy). For each possible split, you calculate the impurity of the resulting child nodes.
3. **Calculate the information gain for each possible split**: The information gain is the impurity of the parent node minus the weighted sum of the impurities of the child nodes. The weights are the proportions of instances that would go to each child node if that split was chosen.
4. **Choose the split with the highest information gain**: The split that results in the highest information gain is the one that reduces the impurity the most, and is therefore the "best" split.

Note: While the question mentions finding the split that induces the largest entropy, this is not correct. We actually want to find the split that reduces the entropy the most (i.e., gives the largest information gain). Similarly, we want to find the split that minimizes the Gini impurity, not maximizes it.

A diagram of a tree

Description automatically generated

### **Ensemble Models:**

Combining Models so that we can make effective predictions of data are -ensemble-based methods

Ensemble models are a very popular technique as they can assist your models be more resistant to outliers and have better chances at generalizing with future data. They also gained popularity after several ensembles helped people win prediction competitions. Recently, stochastic gradient boosting became a go-to candidate model for many data scientists. This model walks you through the theory behind ensemble models and popular tree-based ensembles.

* Identify, use, and interpret common ensemble models for classification, including bagging, boosting, stacking, and random forest.
* Build ensemble models with sklearn, including bagging, boosting, stacking, and random forest.
* Identify common supervised machine learning algorithms.

1. **Ensemble and Bagging: Bootstrap aggregation**

Instead of pruning in decision trees to reduce overfitting. We use many trees and combine aggregation of the output to predict target

**A diagram of a tree

Description automatically generated**

**A diagram of data analysis

Description automatically generated**

**But how to decide how many numbers of trees to be created to do bagging?**

This is decided by a hyperparameters

A graph of a tree

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

**2)Random Forest:**

With bagging we can reduce variance, but we can reduce variance further with Random Forest.

In bagging for n independent trees each with sigma square the bagged variance is (sigma)2/n. as given below

A graph with numbers and lines

Description automatically generated

But it is leads to higher correlated values.

That means some trees are identical to each other, so we need to increase randomness of the trees to decrease correlation or de-correlate it.

Here we also use random columns/features subset for each tree.

Generally for other models we use random rows but for this we use random columns and rows both.

A graph of a number of bags

Description automatically generated

A screenshot of a computer program

Description automatically generated

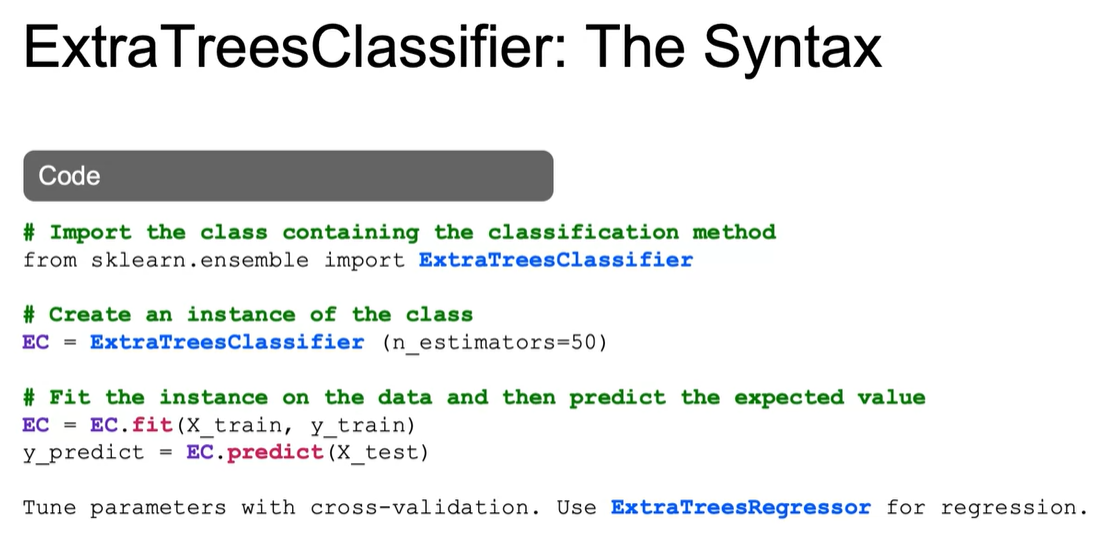
**3)ExtraTreesClassifier:**

Sometimes random Forest randomness is not enough, we may need extra randomness in creating these trees then we use extra tree classifier.

A white background with black text

Description automatically generated

We choose greedily splits in Decision trees, but here we split trees randomly and create trees randomly.



### **Ensemble Based Methods:**

### **Ensemble Based Methods and Bagging**

Tree ensembles have been found to generalize well when scoring new data. Some useful and popular tree ensembles are bagging, boosting, and random forests. Bagging, which combines decision trees by using bootstrap aggregated samples. An advantage specific to bagging is that this method can be multithreaded or computed in parallel. Most of these ensembles are assessed using out-of-bag error.

### **Random Forest**

Random forest is a tree ensemble that has a similar approach to bagging. Their main characteristic is that they add randomness by only using a subset of features to train each split of the trees it trains. Extra Random Trees is an implementation that adds randomness by creating splits at random, instead of using a greedy search to find split variables and split points.

### **Boosting**

Boosting methods are additive in the sense that they sequentially retrain decision trees using the observations with the highest residuals on the previous tree. To do so, observations with a high residual are assigned a higher weight.

### **Gradient Boosting**

The main loss functions for boosting algorithms are:

* 0-1 loss function, which ignores observations that were correctly classified. The shape of this loss function makes it difficult to optimize.
* Adaptive boosting loss function, which has an exponential nature. The shape of this function is more sensitive to outliers.
* Gradient boosting loss function. The most common gradient boosting implementation uses a binomial log-likelihood loss function called deviance. It tends to be more robust to outliers than AdaBoost.

The additive nature of gradient boosting makes it prone to overfitting. This can be addressed using cross validation or fine tuning the number of boosting iterations. Other hyperparameters to fine tune are:

* learning rate (shrinkage)
* subsample
* number of features.

### **Stacking**

Stacking is an ensemble method that combines any type of model by combining the predicted probabilities of classes. In that sense, it is a generalized case of bagging. The two most common ways to combine the predicted probabilities in stacking are: using a majority vote or using weights for each predicted probability.

### **Modeling Unbalanced Classes**

Classification algorithms are built to optimize accuracy, which makes it challenging to create a model when there is not a balance across the number of observations of different classes. Common methods to approach balancing the classes are:

* Downsampling or removing observations from the most common class
* Upsampling or duplicating observations from the rarest class or classes
* A mix of downsampling and upsampling

### **Modeling Approaches for Unbalanced Classes**

Specific algorithms to upsample and downsample are:

* Stratified sampling
* Random oversampling
* Synthetic oversampling, the main two approaches being Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic sampling (ADASYN)
* Cluster Centroids implementations like NearMiss, Tomek Links, and Nearest Neighbors