



Define Regression. Explain types of regression.

Regression is a statistical method used to understand the relationship between a dependent variable (often called the outcome or response variable) and one or more independent variables (often called predictors or explanatory variables). The goal of regression is to model this relationship so that we can predict the dependent variable based on the values of the independent variables, and also understand the strength and nature of these relationships.

Types of Regression

1. Linear Regression

Definition: Linear regression models the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. The simplest form is the simple linear regression, which involves one dependent and one independent variable.

Equation: $Y = \beta_0 + \beta_1 X + \epsilon$

Where:

- Y is the dependent variable.
- β_0 is the intercept.
- β_1 is the slope (coefficient of the independent variable X).
- ϵ is the error term.

Types:

- Simple Linear Regression: One independent variable.
- Multiple Linear Regression: More than one independent variable.

Multiple linear regression

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon$$

where:

- y is the dependent variable.
- x_1, x_2, \dots, x_n are the independent variables.
- β_0 is the y-intercept (constant term).
- $\beta_1, \beta_2, \dots, \beta_n$ are the slopes of the line (regression coefficients).
- ϵ is the error term.

2. Polynomial Regression

Definition: Polynomial regression is a type of linear regression where the relationship between the independent variable X and the dependent variable Y is modeled as an n th degree polynomial.

Equation: $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_n X^n + \epsilon$

Use Case: Useful when the data shows a non-linear relationship that can be approximated by a polynomial.

3. Logistic Regression

Definition: Logistic regression is used when the dependent variable is binary (i.e., it takes two possible outcomes). It models the probability that a given input point belongs to a particular category.

Equation: Sigmoid function

Use Case: Commonly used for classification problems such as spam detection, disease diagnosis, etc.

4. Ridge Regression

Definition: Ridge regression is a type of linear regression that includes a regularization term (also known as L2 regularization) to penalize large coefficients, helping to prevent overfitting.

Equation: $\min \left(\sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}))^2 + \lambda \sum_{j=1}^p \beta_j^2 \right)$

Where λ is the regularization parameter.

Use Case: Useful when there are multicollinearity issues or when the number of predictors is large.

5. Lasso Regression

Definition: Lasso regression (Least Absolute Shrinkage and Selection Operator) is similar to ridge regression but uses L1 regularization. It can shrink some coefficients to zero, effectively performing variable selection.

Equation: $\min \left(\sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}))^2 + \lambda \sum_{j=1}^p |\beta_j| \right)$

Use Case: Useful for feature selection and when we want a sparse model with fewer predictors.

Ridge Regression (L2 norm)

- It is used when there is high correlation between the independent variables
- λ (lambda) solves the problem of Multicollinearity:

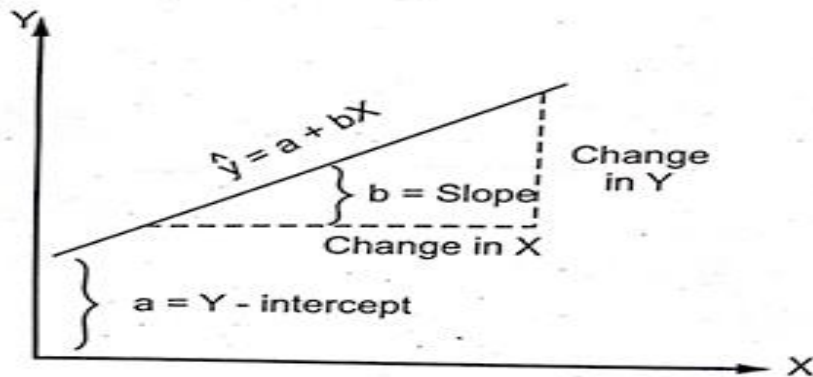
6. Elastic Net Regression

Definition: Elastic Net regression combines both L1 and L2 regularization terms from lasso and ridge regression.

Equation:

$$\min \left(\sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}))^2 + \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^p \beta_j^2 \right)$$

Use Case: Useful when there are multiple correlated predictors and we want to perform feature selection while maintaining regularization.



Univariate Regression

Univariate regression, also known as simple regression, involves one dependent variable and one independent variable. It aims to find a linear relationship between the two variables. The simplest form of univariate regression is linear regression, which can be represented by the equation:

$$y = \beta_0 + \beta_1 x + \epsilon$$

where:

- y is the dependent variable.
- x is the independent variable.
- β_0 is the y-intercept (constant term).
- β_1 is the slope of the line (regression coefficient).
- ϵ is the error term.

Multivariate Regression

Multivariate regression, also known as multiple regression, involves one dependent variable and two or more independent variables. It aims to model the relationship between the dependent variable and several independent variables simultaneously. The general form of multivariate linear regression is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon$$

where:

- y is the dependent variable.
- x_1, x_2, \dots, x_n are the independent variables.
- β_0 is the y-intercept (constant term).
- $\beta_1, \beta_2, \dots, \beta_n$ are the slopes of the line (regression coefficients).
- ϵ is the error term.

Key Differences

- **Number of Independent Variables:** Univariate regression involves one independent variable, while multivariate regression involves two or more independent variables.
- **Complexity:** Univariate regression is simpler and easier to interpret compared to multivariate regression, which can become complex with the addition of more independent variables.
- **Use Cases:** Univariate regression is suitable when the relationship between a single independent variable and the dependent variable is to be studied. Multivariate regression is used when the influence of multiple factors on the dependent variable needs to be understood.

Applications

- **Univariate Regression:** Predicting house prices based on square footage, predicting sales based on advertising expenditure, etc.
- **Multivariate Regression:** Predicting house prices based on square footage, number of bedrooms, and location; predicting sales based on advertising expenditure, market conditions, and competitor actions.

Aspect	Univariate Regression	Multivariate Regression
Definition	Regression with one dependent variable and one independent variable	Regression with one dependent variable and multiple independent variables
Number of Variables	One independent variable	Two or more independent variables
Equation	$y = \beta_0 + \beta_1x + \epsilon$	$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n + \epsilon$
Complexity	Simpler and easier to interpret	More complex, harder to interpret
Use Case	When analyzing the effect of a single variable on the dependent variable	When analyzing the effect of multiple variables on the dependent variable
Computation	Requires fewer computations	Requires more computations
Interpretation	Easier to understand and visualize	More challenging to understand and visualize due to multiple variables
Example	Predicting house price based on square footage	Predicting house price based on square footage, number of bedrooms, and location
Application	Basic predictive modeling and trend analysis	Advanced predictive modeling involving multiple factors
Assumptions	Assumes linear relationship between two variables	Assumes linear relationship between the dependent variable and each independent variable

Linear Regression

Linear regression models the relationship between the dependent variable and one or more independent variables using a linear function.

Characteristics:

- **Equation:** The relationship is expressed as a linear equation:
 - Simple linear regression: $y = \beta_0 + \beta_1 x + \epsilon$
 - Multiple linear regression: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon$
- **Linearity:** Assumes a linear relationship between the dependent and independent variables.
- **Interpretation:** Coefficients (β) represent the change in the dependent variable for a one-unit change in the independent variable.
- **Ease of Computation:** Simpler to compute and interpret.
- **Common Methods:** Ordinary Least Squares (OLS) is the most common method for estimating the coefficients.

Applications:

- Predicting outcomes based on one or more predictor variables (e.g., predicting house prices based on size, location, and number of bedrooms).
- Assessing the strength of predictors.
- Identifying trends and forecasting.



Nonlinear Regression

Nonlinear regression models the relationship between the dependent variable and one or more independent variables using a nonlinear function.

Characteristics:

- **Equation:** The relationship is expressed as a nonlinear equation: $y = f(x, \beta) + \epsilon$, where f is a nonlinear function of the parameters.
- **Nonlinearity:** Assumes a nonlinear relationship between the dependent and independent variables.
- **Flexibility:** Can model more complex relationships compared to linear regression.
- **Interpretation:** Coefficients can be more difficult to interpret directly.
- **Computation:** More complex and computationally intensive. Often requires iterative methods for estimation (e.g., nonlinear least squares, maximum likelihood estimation).

Applications:

- Modeling growth rates (e.g., population growth, tumor growth).
- Analyzing dose-response curves in pharmacokinetics.
- Predicting outcomes in systems where the relationship between variables is inherently nonlinear (e.g., enzyme kinetics in biochemistry).

Aspect	Simple Linear Regression	Multiple Linear Regression
Definition	Models the relationship between one dependent variable and one independent variable	Models the relationship between one dependent variable and two or more independent variables
Equation	$y = \beta_0 + \beta_1 x + \epsilon$	$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon$
Number of Independent Variables	One	Two or more
Complexity	Simpler, easier to compute and interpret	More complex, requires more computations and interpretation
Visualization	Easy to visualize (2D graph)	Harder to visualize (multidimensional space)
Use Cases	Analyzing the effect of a single factor	Analyzing the combined effect of multiple factors
Computation	Requires fewer computational resources	Requires more computational resources
Assumptions	Assumes a linear relationship between two variables	Assumes a linear relationship between the dependent variable and each independent variable
Interpretation	Slope (β_1) represents the change in y for a one-unit change in x	Each slope (β_i) represents the change in y for a one-unit change in x_i , holding other variables constant
Example	Predicting house price based on square footage 	Predicting house price based on square footage, number of bedrooms, and location

The Bias-Variance tradeoff is a fundamental concept in machine learning that deals with the balance between two sources of error that affect the performance of models: bias and variance. Understanding this tradeoff is crucial for developing models that generalize well to new, unseen data.

Bias

- **Definition:** Bias refers to the error introduced by approximating a real-world problem, which may be complex, by a simplified model.
- **High Bias:** Models with high bias are usually too simple and fail to capture the underlying patterns in the data. This results in underfitting, where the model performs poorly both on the training data and on new, unseen data.
- **Examples:** Linear regression models, where a linear model is used to fit data that may have a more complex, nonlinear relationship.

Variance

- **Definition:** Variance refers to the error introduced by the model's sensitivity to small fluctuations in the training data.
- **High Variance:** Models with high variance are usually too complex and capture noise along with the underlying patterns in the data. This results in overfitting, where the model performs well on the training data but poorly on new, unseen data.
- **Examples:** Decision trees with too many branches or high-degree polynomial regression models.

Bias in machine learning is like a systematic error that happens when a model is too simple to capture the true patterns in the data. It leads to the model making consistent mistakes because it doesn't learn the complexities of the data.

For example, imagine trying to predict house prices using only the size of the house. If you ignore other important factors like location or number of bedrooms, your predictions will be consistently off, showing high bias.

Variance in machine learning refers to the error that occurs when a model is too sensitive to the specific details of the training data. It happens when the model is too complex and captures the noise along with the underlying patterns, making it perform well on the training data but poorly on new, unseen data. Think of it as overfitting the problem.

For example, imagine trying to predict house prices and using every single detail about each house, including minor and irrelevant details like the color of the curtains. Your model might fit the training data perfectly but will likely make poor predictions on new data because it learned to fit the noise rather than the actual trend, showing high variance.

The Tradeoff

- **Balancing Bias and Variance:** The goal is to find a balance where the model is complex enough to capture the underlying patterns in the data (low bias) but not so complex that it also captures noise (low variance).
- **Training and Testing Error:** Typically, as the model complexity increases, the training error decreases (because the model fits the training data better), but the testing error (error on new data) first decreases and then starts to increase when the model begins to overfit the training data.

Visualization

1. **Underfitting (High Bias, Low Variance):** The model is too simple. It fails to capture the underlying trend in the data.
2. **Overfitting (Low Bias, High Variance):** The model is too complex. It captures the noise in the training data as if it were a part of the trend.
3. **Optimal Model (Low Bias, Low Variance):** The model captures the underlying trend without fitting the noise in the training data.

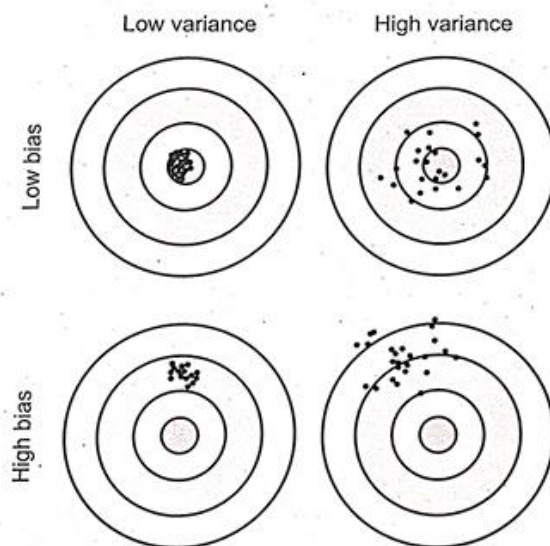


Fig. 2.4.2 Bias-variance trade off

Practical Approaches to Address the Tradeoff

1. **Cross-Validation:** Use techniques like k-fold cross-validation to estimate model performance and adjust the complexity.
2. **Regularization:** Apply regularization techniques (e.g., Lasso, Ridge) to penalize overly complex models.
3. **Ensemble Methods:** Use ensemble methods (e.g., bagging, boosting) to combine multiple models and reduce variance without substantially increasing bias.
4. **Hyperparameter Tuning:** Adjust hyperparameters (e.g., depth of decision trees, degree of polynomial) to find the optimal balance.

In machine learning, underfitting and overfitting are two common problems that can occur when training models. They both relate to the model's performance on training data versus unseen test data.

Underfitting

Underfitting happens when a model is too simple to capture the underlying patterns in the data. This means that the model performs poorly on both the training data and new data. It is usually a sign that the model has high bias.

Characteristics of Underfitting:

- Poor performance on training data and test data.
- High bias.
- Model is too simple, lacking the complexity to capture the underlying trend.

Techniques to Reduce Underfitting:

1. **Increase Model Complexity:** Use a more complex model with more parameters.
2. **Feature Engineering:** Add more features or use more relevant features.
3. **Reduce Regularization:** Decrease the regularization parameters if they are too high.
4. **Increase Training Time:** Train the model for a longer time if using iterative algorithms.

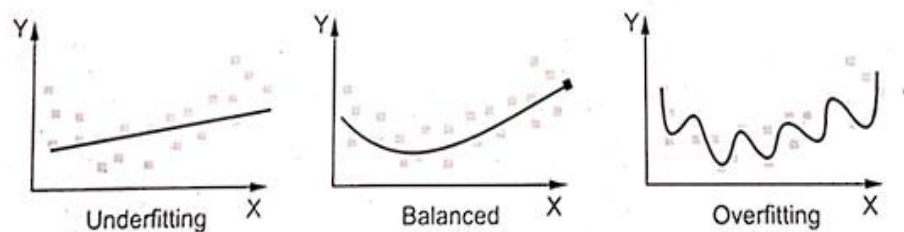


Fig. 2.4.1

Overfitting

Overfitting happens when a model is too complex and captures noise or random fluctuations in the training data instead of the underlying pattern. This means that the model performs very well on the training data but poorly on new, unseen data. It is usually a sign that the model has high variance.

Characteristics of Overfitting:

- Excellent performance on training data but poor performance on test data.
- High variance.
- Model is too complex, fitting to the noise in the training data.

Techniques to Reduce Overfitting:

1. **Simplify the Model:** Use a simpler model with fewer parameters.
2. **Regularization:** Apply techniques like L1 (Lasso) and L2 (Ridge) regularization to penalize large coefficients.
3. **Cross-Validation:** Use techniques like k-fold cross-validation to ensure the model generalizes well to unseen data.
4. **Pruning (for decision trees):** Remove parts of the model that provide little power in predicting target variables.
5. **Early Stopping:** Monitor the model's performance on a validation set and stop training when performance starts to degrade.
6. **Ensemble Methods:** Combine the predictions of multiple models (e.g., bagging, boosting) to reduce overfitting.
7. **Dropout (for neural networks):** Randomly drop units during training to prevent over-reliance on specific neurons.
8. **Data Augmentation:** Increase the size of the training dataset by creating modified versions of the training data.
9. **Train with More Data:** Increasing the amount of training data can help the model to learn the underlying patterns better.

2.11 Evaluation metrics

- Mean Squared Error (MSE), and Mean Absolute Error (MAE) are used to evaluate the regression problem's accuracy.

2.11.1 Mean Squared Error

- Mean Squared Error (MSE) is calculated by taking the average of the square of the difference between the original and predicted values of the data. It can also be called the quadratic cost function or sum of squared errors.

- The value of MSE is always positive or greater than zero. A value close to zero will represent better quality of the estimator/predictor. An MSE of zero (0) represents the fact that the predictor is a perfect predictor.

$$MSE = \frac{1}{N} \sum_{i=1}^n (\text{Actual values} - \text{Predicted values})^2$$

- Here N is the total number of observations/rows in the dataset. The sigma symbol denotes that the difference between actual and predicted values taken on every i value ranging from 1 to n.

- Mean squared error is the most commonly used loss function for regression. MSE is sensitive towards outliers and given several

examples with the same input feature values, the optimal prediction will be their mean target value. This should be compared with Mean Absolute Error, where the optimal prediction is the median. MSE is thus good to use if you believe that your target data, conditioned on the input, is normally distributed around a mean value, and when it's important to penalize outliers extra much.

- MSE incorporates both the variance and the bias of the predictor. MSE also gives more weight to larger differences. The bigger the error, the more it is penalized.
- Example : You want to predict future house prices. The price is a continuous value, and therefore we want to do regression. MSE can here be used as the loss function.

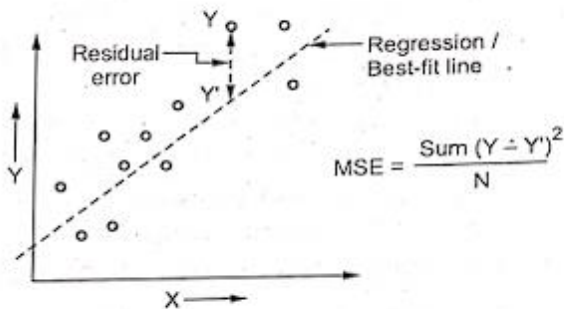


Fig. 2.11.1 Representation of MSE

2. Mean Absolute Error (MAE)

- **Definition:** MAE measures the average absolute difference between the actual and predicted values. It's calculated as:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- **Interpretation:** MAE is easier to interpret than MSE because it is in the same unit as the data. It gives a linear penalty for errors, meaning each error contributes equally.

- Use mean absolute error when you are doing regression and don't want outliers to play a big role. It can also be useful if you know that your distribution is multimodal. MAE loss is useful if the training data is corrupted with outliers.

3. Root Mean Squared Error (RMSE)

- **Definition:** RMSE is the square root of the MSE. It's calculated as:

$$\text{RMSE} = \sqrt{\text{MSE}}$$

- **Interpretation:** RMSE is also in the same unit as the data and provides a measure of how spread out the errors are. It tends to give more weight to larger errors due to the squaring process.

- Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are.

RMSE is always non-negative and a value of 0 would indicate a perfect fit to the data

RMSE is sensitive to outliers

4. R-squared (R^2)

- **Definition:** R^2 measures the proportion of the variance in the dependent variable that is predictable from the independent variables. It's calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

where \bar{y} is the mean of the actual values.

- **Interpretation:** R^2 values range from 0 to 1, with higher values indicating a better fit. An R^2 of 1 means the model perfectly predicts the outcomes.

- R-squared can also be expressed as a function of mean squared error. R-squared represents the fraction of variance of response variable captured by the regression model rather than the MSE which captures the residual error.

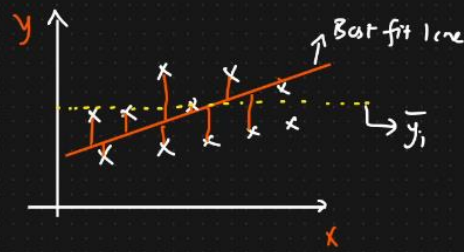
$$100\% \left\{ \text{overfitting} \right\} = 1 - \frac{\text{Small number}}{\text{Big number}} \Rightarrow \text{Small number}$$

$0.25 = 0.25 = 25\%$

① R squared

$$= 1 - \frac{SS_{Res}}{SS_{Total}}$$

$$= 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$



The R^2 value varies between 0 and 1 where 0 represents no correlation between the predicted and actual value and 1 represents complete correlation.

R-squared is a good measure to evaluate the model fitness. It is also known as the coefficient of determination. R-squared is the fraction by which the variance of the errors is less than the variance of the dependent variable.

It is called R-squared because in a simple regression model it is just the square of the correlation between the dependent and independent variables, which is commonly denoted by "r".

5. Adjusted R-squared

- **Definition:** Adjusted R^2 adjusts R^2 for the number of predictors in the model. It's calculated as:

N = No. of datapoints
 P = No. of independent features

$$\text{Adjusted } R^2 = 1 - \left(\frac{1 - R^2}{n - p - 1} \right) \times (n - 1)$$

where p is the number of predictors and n is the number of observations.

- **Interpretation:** Adjusted R^2 is useful for comparing models with different numbers of predictors. It penalizes the addition of predictors that do not improve the model, helping to prevent overfitting.

- Adjusted R^2 is always less than or equal to R^2 . A value of 1 indicates a model that perfectly predicts values in the target field. A value that is less than or equal to 0 indicates a model that has no predictive value. In the real world, adjusted R^2 lies between these values.
- Adjusted R-squared can be negative when R-squared is close to zero. Adjusted R-squared value always be less than or equal to R-squared value.

Polynomial regression is an extension of linear regression that allows for modeling the relationship between the independent variable x and the dependent variable y as an n -degree polynomial. While linear regression assumes a straight-line relationship, polynomial regression can fit a curve to the data, making it more flexible for capturing non-linear patterns.

Key Concepts

1. **Polynomial Features:** In polynomial regression, we transform the original features by raising them to various powers. For example, if you have a feature x , a polynomial regression of degree 2 would include features x and x^2 .
2. **Model Representation:** The polynomial regression model can be represented as:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n + \epsilon$$

where $\beta_0, \beta_1, \dots, \beta_n$ are the coefficients, and n is the degree of the polynomial.

3. **Fitting the Model:** The coefficients β are estimated using methods such as Ordinary Least Squares (OLS), similar to linear regression. The model aims to minimize the difference between the predicted values and the actual values.
4. **Choosing the Degree:** The degree of the polynomial is a critical choice. A higher-degree polynomial can fit the training data better but may lead to overfitting. A lower-degree polynomial might underfit the data if it's too simple.
5. **Regularization:** To address the risk of overfitting, techniques like regularization (e.g., Ridge or Lasso regression) can be applied. Regularization adds a penalty term to the cost function to constrain the magnitude of the coefficients.

Advantages

- **Flexibility:** Can model complex relationships between variables.
- **Better Fit for Non-linear Data:** Captures trends that linear regression might miss.

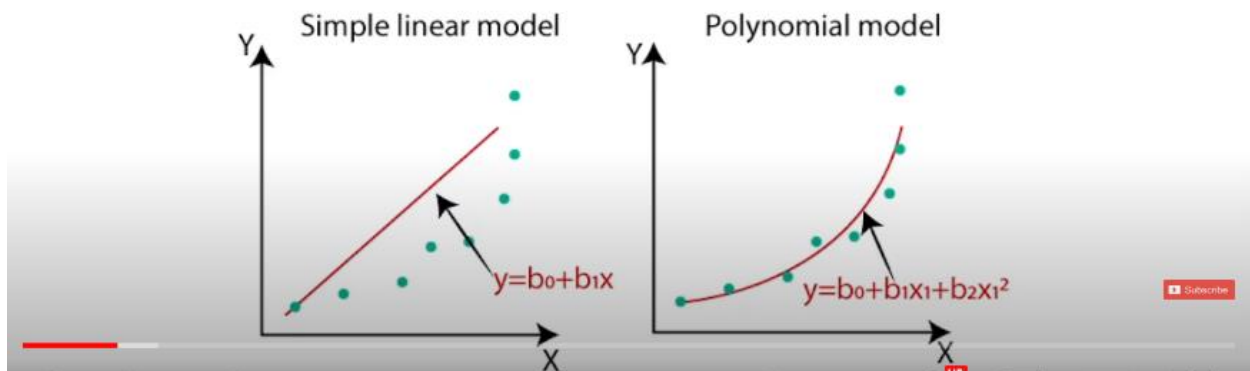
Disadvantages

- **Overfitting:** High-degree polynomials can overfit the training data and perform poorly on unseen data.
- **Computational Complexity:** Higher-degree polynomials can increase computational costs and complexity.

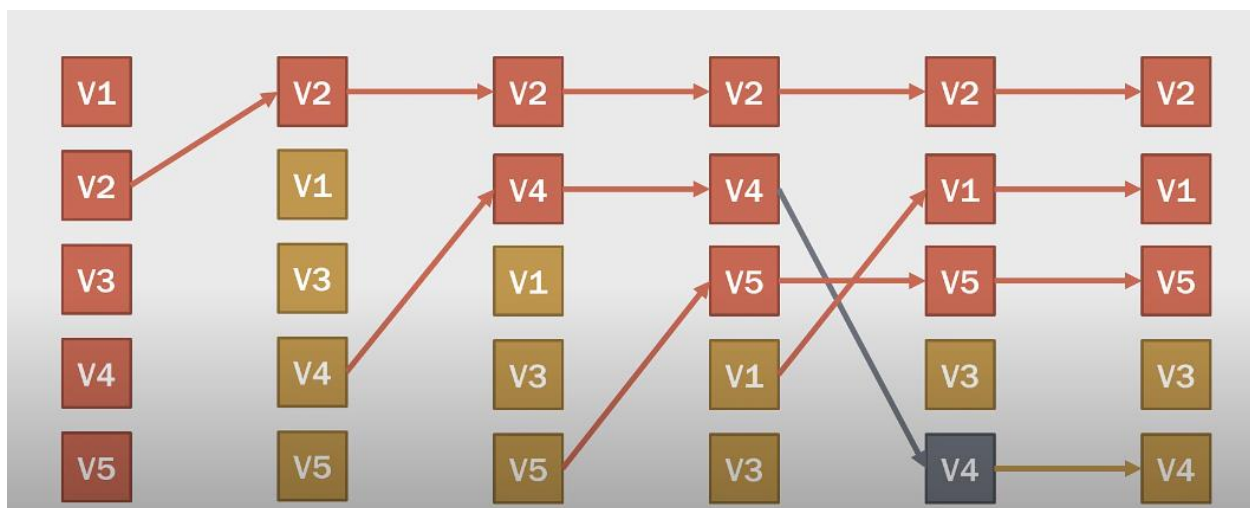
Example

Suppose you have a dataset with x and y values, and you want to fit a polynomial curve:

1. **Feature Transformation:** Convert x to polynomial features (e.g., x^2 , x^3).
2. **Model Training:** Fit a polynomial regression model using these features.
3. **Prediction:** Use the fitted model to make predictions on new data.



Stepwise process



- Stepwise is just a combination of forward selection and backward elimination
- There are two threshold values: entry and exit
- Usually, threshold to exit is set more liberally; for example, 0.05 to enter and 0.10 to exit. This has a stability effect, so feature are not flying in and out of the model.
- Forward → Evaluate → Backward → Evaluate... hence, STEPWISE
- At each step, we evaluate the model. If a feature is no longer contributing to reduction of SSE, it is deleted. Then we move forward again.
- Features CAN reenter at a later step as the model evolves



Stepwise regression is a systematic method for selecting the most relevant variables in a regression model. It involves adding or removing predictors to find a model that best balances complexity and performance. There are three main types of stepwise regression:

1. Forward Selection

- **Start with No Predictors:** Begin with an empty model.
- **Add Predictors:** Iteratively add predictors one by one. At each step, add the predictor that improves the model the most (e.g., the one with the lowest p-value or highest improvement in R^2).
- **Stop Criterion:** Continue adding predictors until no further improvement is observed or adding more predictors starts degrading model performance.

2. Backward Elimination

- **Start with All Predictors:** Begin with a model that includes all candidate predictors.
- **Remove Predictors:** Iteratively remove the least significant predictor (e.g., the one with the highest p-value) that does not significantly affect the model performance.
- **Stop Criterion:** Continue removing predictors until only significant variables remain or removing further predictors starts worsening the model.

3. Bidirectional (Stepwise) Selection

- **Combine Forward and Backward Methods:** Start with an empty model or a full model and alternately add and remove predictors.
- **Add and Remove:** At each step, evaluate the addition of new predictors as well as the removal of existing ones. This process helps to find a balance between model complexity and fit.
- **Stop Criterion:** Continue until no further improvements can be made by adding or removing predictors.

Criteria for Variable Selection

- **P-value:** A common criterion is the p-value associated with each predictor. Predictors with p-values below a certain threshold (e.g., 0.05) are considered significant.
- **AIC/BIC:** Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are used to balance model fit and complexity. Lower values of AIC/BIC indicate a better model.
- **Adjusted R^2 :** Adjusted R^2 accounts for the number of predictors in the model, providing a measure of fit that penalizes excessive predictors.

Criteria for Variable Selection

- **P-value:** A common criterion is the p-value associated with each predictor. Predictors with p-values below a certain threshold (e.g., 0.05) are considered significant.

- **Adjusted R^2 :** Adjusted R^2 accounts for the number of predictors in the model, providing a measure of fit that penalizes excessive predictors.

Advantages

- **Automatic Variable Selection:** Simplifies the model selection process by automating the choice of predictors.
- **Improves Model Interpretability:** By reducing the number of predictors, the model becomes easier to interpret.

Disadvantages

- **Overfitting Risk:** There's a risk of overfitting if the method selects too many predictors or if the data is noisy.
- **Computational Cost:** Especially for large datasets with many predictors, stepwise regression can be computationally intensive.
- **Model Stability:** The selected model might not be stable across different samples of the data due to its reliance on data-specific properties.

Decision Tree Regression is a type of regression model that uses a decision tree structure to predict a continuous target variable. Unlike linear regression, which models relationships as linear equations, decision tree regression models complex, non-linear relationships by partitioning the feature space into regions and assigning a value to each region.

Key Concepts

1. Tree Structure:

- **Nodes:** The tree consists of nodes where each node represents a feature or a decision rule.
- **Branches:** The branches represent the outcome of a decision rule that splits the data into subsets.
- **Leaves:** The terminal nodes or leaves of the tree contain the predicted value for the target variable.

2. Splitting Criteria:

- At each internal node, the tree uses a criterion to determine the best feature and threshold for splitting the data. Common criteria include:
 - **Mean Squared Error (MSE):** Measures the average squared difference between the predicted values and the actual values. The split that minimizes MSE is preferred.
 - **Mean Absolute Error (MAE):** Measures the average absolute difference between the predicted values and the actual values.

3. Recursive Partitioning:

- The tree is built recursively by splitting the data at each node based on the chosen criterion. This process continues until a stopping condition is met, such as a maximum tree depth or minimum number of samples in a node.

4. Overfitting:

- Decision trees can easily overfit the training data, especially if they grow too deep. Overfitting occurs when the tree captures noise or anomalies in the training data, leading to poor generalization to new data.

5. Pruning:

- Pruning is a technique used to reduce the size of the tree and prevent overfitting. It involves removing nodes or branches that have little importance or contribute minimally to the model's performance.

Advantages

- **Non-Linearity:** Can model complex, non-linear relationships between features and the target variable.
- **Interpretability:** Easy to visualize and interpret. The decision-making process is transparent and can be understood through the tree structure.
- **No Feature Scaling:** Does not require normalization or scaling of features.

Disadvantages

- **Overfitting:** Prone to overfitting, especially with deep trees and noisy data.
- **Instability:** Small changes in the data can lead to a significantly different tree structure.
- **Bias:** Can be biased towards features with more levels or categories.

Example

Suppose you want to predict house prices based on features like square footage, number of bedrooms, and location:

1. **Build the Tree:** Start with the entire dataset and split it based on the feature that best separates the data (e.g., square footage). Continue splitting recursively based on other features until a stopping criterion is met.
2. **Prediction:** For a new house, traverse the tree using the feature values of the house to reach a leaf node, which provides the predicted price.

Practical Considerations

- **Ensemble Methods:** Decision trees are often used as base models in ensemble methods like Random Forest and Gradient Boosting to improve performance and reduce overfitting.
- **Hyperparameters:** Important hyperparameters include tree depth, minimum samples per leaf, and minimum samples per split. Tuning these can significantly impact model performance.

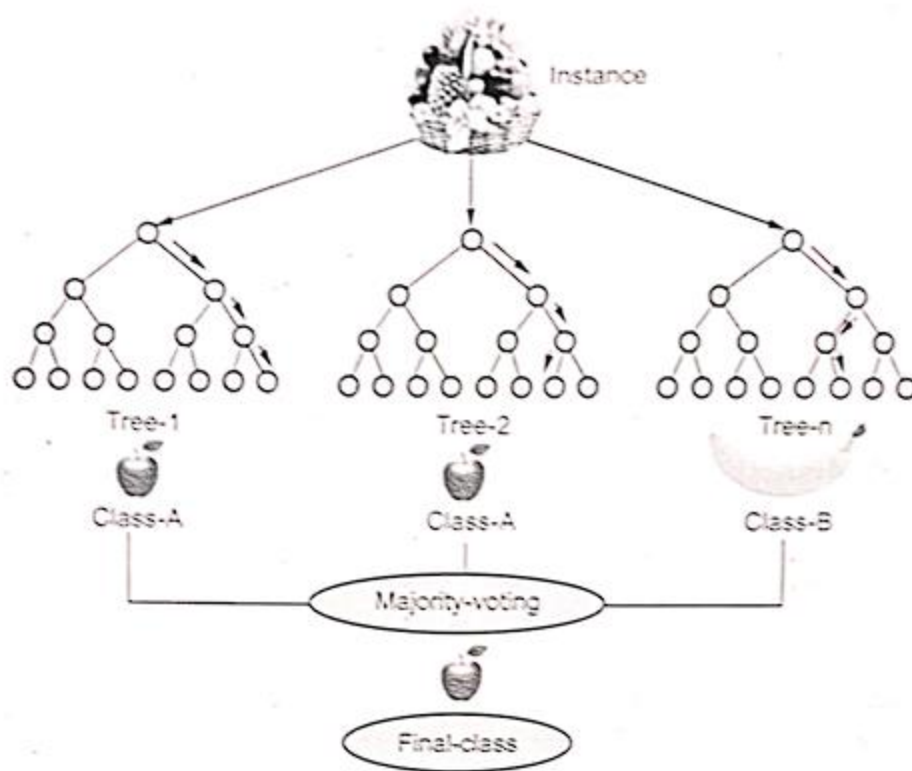
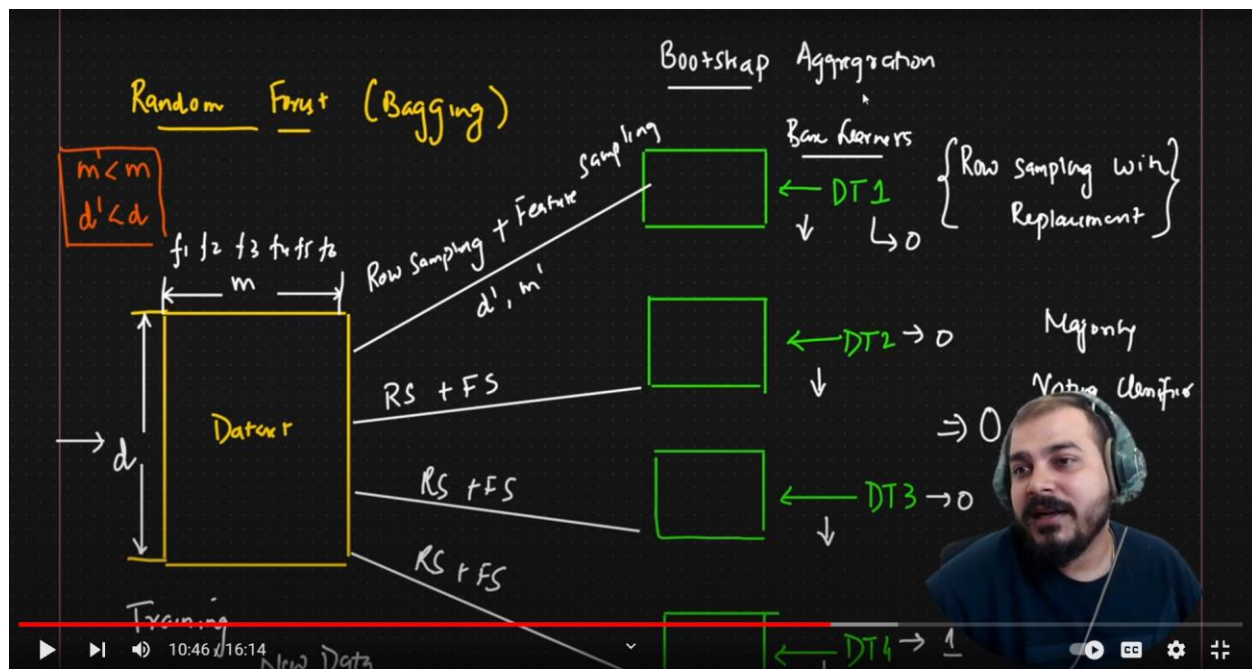


Fig. 2.5.2 Example of random forest



Regression \rightarrow Average \rightarrow Op

D.T

Random Forest

Low Bias \rightarrow Low Bias

High Variance \rightarrow Low Variance

Random Forest Regression is an ensemble learning technique that uses a collection of decision trees to improve prediction accuracy and robustness. It is an extension of decision tree regression that addresses some of the limitations of individual decision trees, such as overfitting and instability.

Key Concepts

1. Ensemble Method:

- **Aggregation:** Random Forest Regression combines the predictions from multiple decision trees to make a final prediction. The final prediction is typically the average of the predictions from all the trees.
- **Bagging:** Short for Bootstrap Aggregating, this technique involves training each decision tree on a different random subset of the training data. Each subset is created by sampling with replacement from the original dataset.

2. Decision Trees:

- **Training:** Each tree in the forest is trained on a different subset of the data, and features are selected randomly at each split. This randomness helps to create diverse trees that contribute to better generalization.
- **Prediction:** For regression tasks, each tree in the forest makes a prediction, and the final output is the average of these predictions.

3. Feature Randomness:

- **Random Feature Selection:** At each node, only a random subset of features is considered for splitting. This further enhances the diversity among trees and helps prevent overfitting.

4. Advantages:

- **Robustness:** Random Forests are less prone to overfitting compared to individual decision trees because they average out the predictions of multiple trees.
- **Feature Importance:** The model can provide insights into the importance of different features for the prediction task.
- **Handling Missing Values:** Can handle missing values better and can be robust to noisy data.

5. Disadvantages:

- **Complexity:** The model can become quite complex with many trees, making it harder to interpret compared to a single decision tree.
- **Computational Cost:** Training and making predictions with a large number of trees can be computationally intensive.

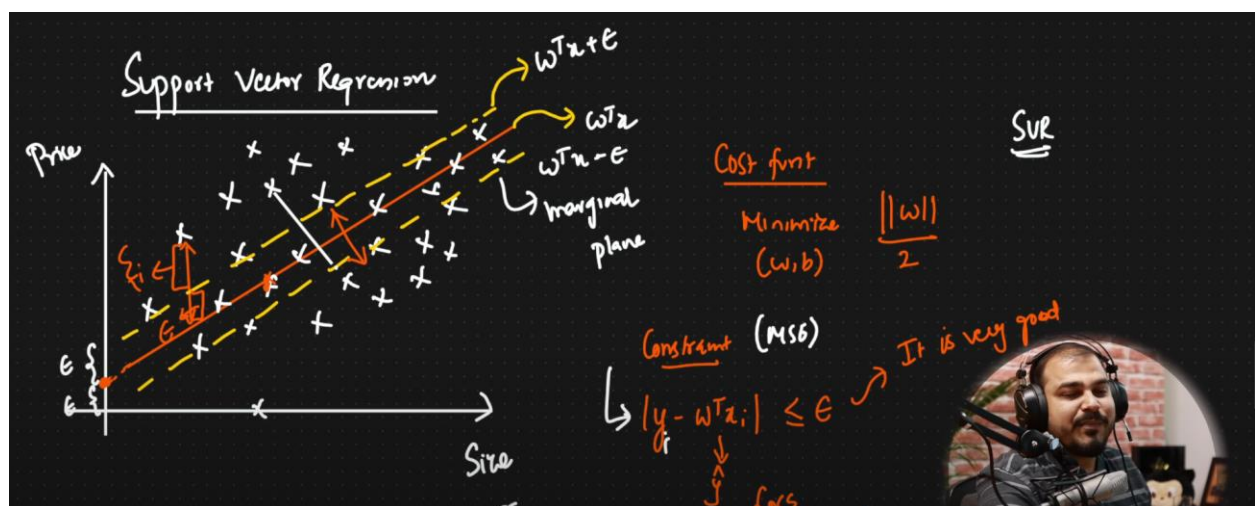
Example

Suppose you want to predict house prices based on features like square footage, number of bedrooms, and location:

1. **Train Multiple Trees:** Create several decision trees, each trained on a different random subset of the data. At each node, only a random subset of features is considered for splitting.
2. **Make Predictions:** For a new house, each tree makes a prediction based on its structure. The final prediction is the average of the predictions from all the trees.

Practical Considerations

- **Hyperparameters:**
 - **Number of Trees ('n_estimators'):** The number of trees in the forest. More trees generally lead to better performance but increase computational cost.
 - **Maximum Depth ('max_depth'):** The maximum depth of each tree. Limiting depth helps control overfitting.
 - **Minimum Samples per Leaf ('min_samples_leaf'):** The minimum number of samples required to be at a leaf node. Increasing this value can prevent overfitting.
 - **Number of Features ('max_features'):** The number of features to consider for splitting at each node. Lower values can increase tree diversity.
- **Feature Importance:** Random Forests can provide a measure of feature importance, which can be useful for understanding which features are most influential in making predictions.



Constraint

$$\underline{|y_i - w_i x_i| \leq \epsilon + \xi_i}$$



Support Vector Regression (SVR) is a type of regression algorithm that uses the principles of Support Vector Machines (SVMs) to predict a continuous target variable. SVR is effective for both linear and non-linear regression tasks and focuses on finding a function that approximates the target values within a certain margin of tolerance.

Key Concepts

1. Hyperplane and Support Vectors:

- **Hyperplane:** In SVR, the goal is to find a hyperplane that best fits the data. For a linear regression problem, this hyperplane is a line.
- **Support Vectors:** These are the data points that lie on the boundary of the margin and are used to define the hyperplane. Only these points influence the position and orientation of the hyperplane.

2. Margin of Tolerance (ϵ -insensitive zone):

- **ϵ -tube:** This is the region around the hyperplane where errors are tolerated. Predictions that fall within this tube are not penalized.
- **Slack Variables:** These variables allow some points to fall outside the ϵ -tube, introducing a penalty proportional to the distance from the tube.

3. Objective Function:

- The objective is to minimize the error, but only those errors that fall outside the ϵ -tube are considered. The optimization problem balances minimizing the error and maximizing the margin.

4. Kernel Trick:

- SVR can handle non-linear relationships using kernel functions that map the input features into a higher-dimensional space where a linear hyperplane can be used for regression.
- Common kernels include linear, polynomial, radial basis function (RBF), and sigmoid.

2.6.3 Limitations of SVM

1. It is sensitive to noise.
2. The biggest limitation of SVM lies in the choice of the kernel.
3. Another limitation is speed and size.
4. The optimal design for multiclass SVM classifiers is also a research area.

Ridge Regression, also known as Tikhonov regularization, is a technique used to analyze multiple regression data that suffer from multicollinearity. It addresses some of the limitations of ordinary least squares (OLS) regression by introducing a regularization term to the cost function, which helps to prevent overfitting and improve the model's generalization.

Key Concepts

1. Ordinary Least Squares (OLS) Regression:

- In OLS, the goal is to minimize the sum of the squared residuals (differences between observed and predicted values):

$$J(\beta) = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2$$

- When predictors are highly correlated (multicollinear), OLS estimates can become unstable, leading to high variance and overfitting.

2. Regularization:

- Regularization adds a penalty to the cost function to constrain the magnitude of the coefficients. In ridge regression, this penalty is the L2 norm of the coefficients.

- The ridge regression cost function is:

$$J(\beta) = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2 + \lambda \sum_{j=1}^p \beta_j^2$$

- Here, λ is the regularization parameter that controls the strength of the penalty. A larger λ increases the penalty on the coefficients, leading to smaller coefficient estimates.

3. Bias-Variance Trade-off:

- Introducing the regularization term reduces the variance of the coefficient estimates but increases their bias. The goal is to find an optimal value of λ that minimizes the overall prediction error.

Advantages

- **Reduces Overfitting:** By adding a penalty to the coefficients, ridge regression prevents them from taking large values, which helps reduce overfitting.
- **Stabilizes Coefficient Estimates:** Particularly useful when dealing with multicollinear data, as it reduces the sensitivity of the coefficients to small changes in the data.
- **Improves Generalization:** Often leads to better out-of-sample prediction performance compared to OLS regression.

Disadvantages

- **Bias Introduction:** Regularization introduces bias into the model, which can sometimes lead to underfitting if λ is too large.
- **Interpretability:** The coefficients are shrunk toward zero, which can make them harder to interpret compared to OLS regression coefficients.

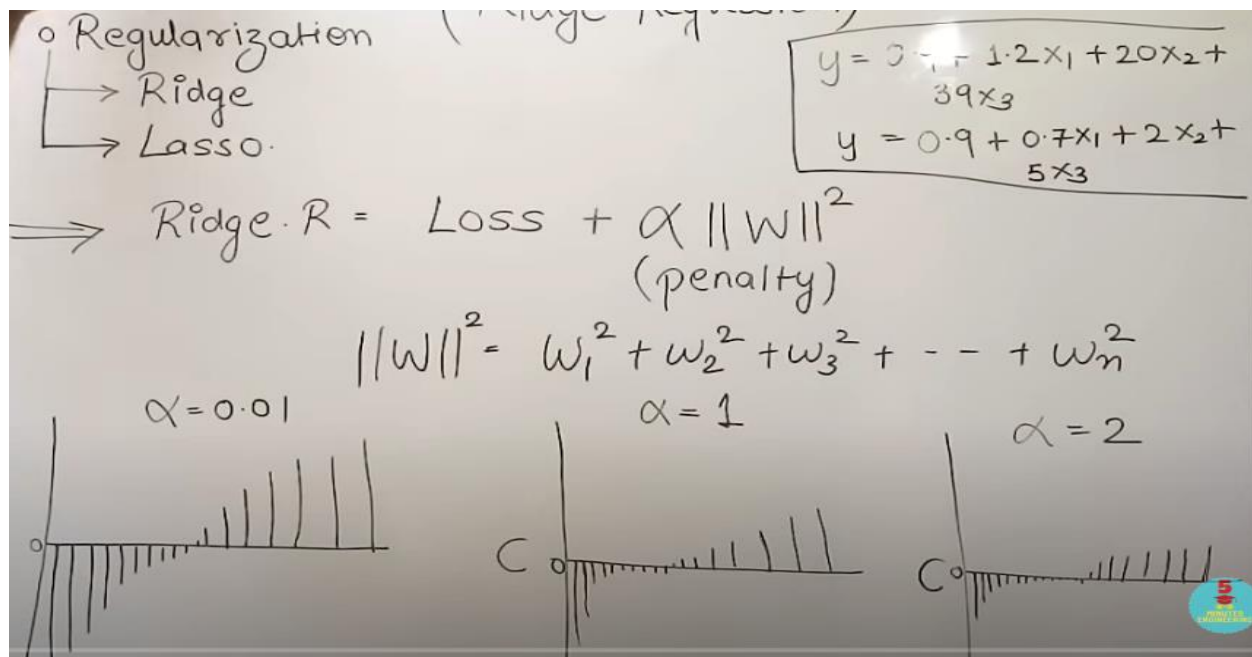
Example

Suppose you have a dataset with several features and you want to predict a continuous target variable:

1. **Model Training:** Fit the ridge regression model to your training data by minimizing the regularized cost function. This involves selecting an appropriate value for λ .
2. **Hyperparameter Tuning:** Use techniques like cross-validation to find the optimal λ that provides the best trade-off between bias and variance.
3. **Prediction:** Use the trained model to make predictions on new data.

Practical Considerations

- **Standardization:** It is often important to standardize the predictors (mean 0, variance 1) before fitting a ridge regression model, as the penalty term depends on the scale of the predictors.
- **Lambda Selection:** Use cross-validation to choose the optimal λ . Common methods include k-fold cross-validation or using a validation set.



Lasso Regression (Least Absolute Shrinkage and Selection Operator) is a type of linear regression that uses L1 regularization to enhance the prediction accuracy and interpretability of the statistical model it produces. Lasso Regression can be particularly useful when dealing with datasets that have a large number of features, many of which might be irrelevant or redundant.

Key Concepts

1. L1 Regularization:

- In Lasso Regression, the cost function includes a penalty term that is the sum of the absolute values of the coefficients. This penalty encourages the model to shrink some coefficients exactly to zero, effectively performing feature selection.
- The cost function for Lasso Regression is:

$$J(\beta) = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- Here, λ is the regularization parameter that controls the strength of the penalty. A larger λ leads to more coefficients being shrunk to zero.

2. Feature Selection:

- One of the key benefits of Lasso Regression is its ability to perform automatic feature selection. As λ increases, more coefficients are driven to zero, leading to a sparse model that only includes the most relevant features.

3. Bias-Variance Trade-off:

- Similar to Ridge Regression, Lasso introduces bias into the model but reduces variance. The goal is to find the optimal value of λ that minimizes the overall prediction error by balancing bias and variance.

Advantages

- **Feature Selection:** By shrinking some coefficients to zero, Lasso Regression effectively selects a simpler model that only includes the most important features.
- **Improves Interpretability:** The resulting model is often easier to interpret because it includes fewer features.
- **Reduces Overfitting:** The regularization term helps to prevent overfitting, especially in datasets with a large number of features.

Disadvantages

- **Bias Introduction:** Lasso introduces bias into the model, which can lead to underfitting if λ is too large.
- **Computational Complexity:** For very large datasets, solving the Lasso optimization problem can be computationally intensive.
- **Limitation with Grouped Variables:** Lasso may not perform well when there are highly correlated features. It tends to select only one feature from a group of correlated features and ignore the others.

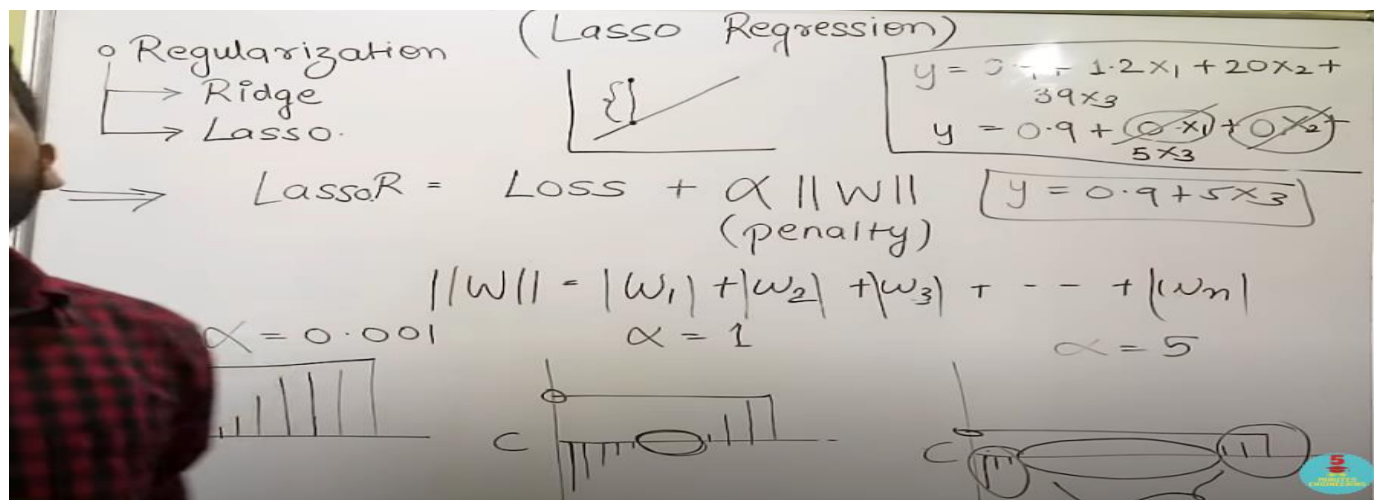
Example

Suppose you want to predict house prices based on features like square footage, number of bedrooms, and location:

1. **Model Training:** Fit the Lasso Regression model to your training data by minimizing the L1-regularized cost function. This involves selecting an appropriate value for λ .
2. **Hyperparameter Tuning:** Use techniques like cross-validation to find the optimal λ that provides the best trade-off between bias and variance.
3. **Prediction:** Use the trained model to make predictions on new data.

Practical Considerations

- **Standardization:** Standardize the predictors (mean 0, variance 1) before fitting a Lasso model, as the penalty term depends on the scale of the predictors.
- **Lambda Selection:** Use cross-validation to choose the optimal λ . Common methods include k-fold cross-validation or using a validation set.
- **Comparison with Ridge Regression:** While Ridge Regression (L2 regularization) shrinks coefficients but does not set any of them to zero, Lasso Regression (L1 regularization) can produce sparse models by setting some coefficients exactly to zero.



ElasticNet Regression is a regularization technique that combines the properties of both Ridge Regression (L2 regularization) and Lasso Regression (L1 regularization). It is particularly useful when dealing with datasets that have many features, especially when those features are correlated.

Key Concepts

1. Combination of L1 and L2 Regularization:

- ElasticNet includes both the L1 and L2 penalty terms in the cost function. The cost function for ElasticNet is:

$$J(\beta) = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2 + \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^p \beta_j^2$$

- Here, λ_1 and λ_2 are the regularization parameters that control the strength of the L1 and L2 penalties, respectively.

2. Feature Selection and Shrinkage:

- The L1 penalty encourages sparsity, leading to feature selection by shrinking some coefficients to zero.
- The L2 penalty helps to handle multicollinearity by spreading the coefficient values among correlated features.

3. Hyperparameters:

- Alpha (α):** The overall regularization strength. It controls the combined effect of L1 and L2 regularization.
- L1 Ratio (ρ):** The mixing parameter that defines the ratio of L1 and L2 regularization. When $\rho = 0$, the penalty is purely L2 (Ridge Regression), and when $\rho = 1$, the penalty is purely L1 (Lasso Regression).

Advantages

- Flexibility:** Combines the benefits of Ridge and Lasso regression, providing a balance between feature selection and coefficient shrinkage.
- Improves Performance:** Can outperform Ridge and Lasso when features are correlated, leading to better predictive performance.
- Handles Multicollinearity:** The L2 component helps to mitigate issues related to multicollinearity.

Disadvantages

- Computational Complexity:** Solving the optimization problem can be computationally intensive, especially for large datasets.
- Hyperparameter Tuning:** Requires careful tuning of both α and ρ to achieve optimal performance.

Example

Suppose you want to predict house prices based on features like square footage, number of bedrooms, and location:

1. **Model Training:** Fit the ElasticNet model to your training data by minimizing the combined L1 and L2-regularized cost function.
2. **Hyperparameter Tuning:** Use techniques like cross-validation to find the optimal values for α and ρ that provide the best trade-off between bias and variance.
3. **Prediction:** Use the trained model to make predictions on new data.

Practical Considerations

- **Standardization:** Standardize the predictors (mean 0, variance 1) before fitting an ElasticNet model, as the penalty terms depend on the scale of the predictors.
- **Alpha and L1 Ratio Selection:** Use cross-validation to choose the optimal values for α and ρ . Common methods include k-fold cross-validation or using a validation set.
- **Implementation:** Many machine learning libraries, such as scikit-learn in Python, provide built-in functions to implement ElasticNet Regression, making it easier to apply and tune.

Bayesian Linear Regression is a statistical method that incorporates Bayesian principles into the linear regression model. Unlike traditional linear regression, which estimates point values for the coefficients, Bayesian linear regression estimates the probability distribution of the coefficients, providing a more comprehensive understanding of the uncertainty in the model.

Key Concepts

1. Bayesian Inference:

- **Prior Distribution:** Represents the initial beliefs about the parameters before observing the data. Common choices for priors in Bayesian linear regression include normal distributions.
- **Likelihood:** The probability of the observed data given the parameters. In linear regression, this is typically modeled as a Gaussian distribution.
- **Posterior Distribution:** Combines the prior distribution and the likelihood of the observed data using Bayes' theorem:

$$p(\beta|X, y) = \frac{p(y|X, \beta)p(\beta)}{p(y|X)}$$

Where $p(\beta|X, y)$ is the posterior distribution, $p(y|X, \beta)$ is the likelihood, $p(\beta)$ is the prior, and $p(y|X)$ is the marginal likelihood.


2. Model:

- The linear model remains the same as in traditional linear regression:

$$y = X\beta + \epsilon$$

Where y is the target variable, X is the matrix of input features, β is the vector of coefficients, and ϵ is the error term, typically assumed to be normally distributed.

3. Posterior Calculation:

- In Bayesian linear regression, we update our beliefs about the parameters β based on the observed data. The posterior distribution  is usually also Gaussian, given Gaussian priors and likelihoods.

Advantages

- **Incorporates Prior Knowledge:** Allows the inclusion of prior information about the parameters.
- **Quantifies Uncertainty:** Provides a full distribution for the coefficients and predictions, giving a measure of uncertainty.
- **Robustness:** Can be more robust to overfitting, especially with appropriate priors.

Disadvantages

- **Computational Complexity:** Can be computationally intensive, especially for large datasets.
- **Choice of Priors:** The results can be sensitive to the choice of prior distributions.

Example

Suppose you want to predict house prices based on features like square footage, number of bedrooms, and location:

1. **Define Priors:** Set prior distributions for the coefficients based on domain knowledge or empirical data.
2. **Fit Model:** Use Bayesian updating to fit the model to your training data, calculating the posterior distribution of the coefficients.
3. **Make Predictions:** Use the posterior distribution to make predictions on new data, quantifying the uncertainty in these predictions.

Practical Considerations

- **Priors Selection:** Choose priors carefully based on prior knowledge or empirical evidence.
- **Computational Methods:** For complex models or large datasets, numerical methods such as Markov Chain Monte Carlo (MCMC) may be used to approximate the posterior distributions.
- **Software:** Various libraries in Python, such as PyMC3, Stan, and scikit-learn's `BayesianRidge`, can be used to implement Bayesian linear regression.

Regression is a fundamental concept in machine learning, especially when the goal is to predict continuous outcomes. Here are several reasons why regression is essential in machine learning:

1. Predicting Continuous Values

- **Use Case:** Regression is used when the target variable is continuous and not categorical. For example, predicting house prices, stock market values, or the temperature.
- **Example:** A model predicting a person's weight based on their height and age.

2. Understanding Relationships Between Variables

- **Use Case:** It helps in understanding the relationships between independent variables (features) and the dependent variable (target).
- **Example:** Understanding how factors like hours studied and attendance affect exam scores.

3. Forecasting

- **Use Case:** Regression is extensively used in time series forecasting, such as predicting future sales, weather forecasting, and economic forecasting.
- **Example:** Predicting next month's sales based on historical sales data.

4. Basis for More Complex Models

- **Use Case:** Many advanced machine learning algorithms build upon regression concepts. Linear regression is often a starting point for understanding more complex models.
- **Example:** Polynomial regression, Ridge regression, Lasso regression, and even some neural network architectures.

5. Feature Impact Analysis

- **Use Case:** Regression models can help in identifying which features have the most significant impact on the target variable.
- **Example:** Determining the key factors that influence customer satisfaction scores.

6. Medical and Biological Applications

- **Use Case:** In medical research, regression models are used to predict patient outcomes based on various health metrics.
- **Example:** Predicting the progression of a disease based on patient data.

7. Risk Management

- **Use Case:** Financial institutions use regression to assess and predict risks, such as credit scoring and loan default probabilities.
- **Example:** Predicting the probability of a borrower defaulting on a loan based on their financial history.

Aspect	Regression	Correlation
Definition	A statistical method to model and analyze the relationship between a dependent variable and one or more independent variables.	A statistical measure that expresses the extent to which two variables are linearly related.
Purpose	Predicts the value of the dependent variable based on the independent variables.	Measures the strength and direction of the relationship between two variables.
Direction of Analysis	Analyzes how the independent variable affects the dependent variable.	Measures the degree of association between two variables without implying causation.
Output	A mathematical equation (regression equation) representing the relationship.	A correlation coefficient (ranging from -1 to 1).
Types	Linear Regression, Multiple Regression, Polynomial Regression, etc.	Pearson Correlation, Spearman Rank Correlation, Kendall Tau Correlation, etc.
Interpretation	Slope and intercept indicate the change in the dependent variable for a unit change in the independent variable.	Correlation coefficient indicates the strength and direction of the relationship.
Causality	Can imply causation if certain conditions are met.	Does not imply causation, only association.
Units of Measurement	Units of the dependent variable.	Dimensionless (unitless).
Range of Values	No specific range.	Between -1 and 1.
Use Cases	Predictive modeling, trend analysis, forecasting, etc.	Measuring relationships, verifying hypotheses, etc.
Graphical Representation	Regression Line (on a scatter plot).	Scatter plot with correlation coefficient.