**PROSIT – 2**

**KEYWORDS a**

* Data processing - Data processing is the method of collecting raw data and translating it into usable information. The raw data is collected, filtered, sorted, processed, analysed, stored, and then presented in a readable format.
* Gradient descent - Gradient descent is an optimization algorithm that is used when training a machine learning model. It is based on a convex function and tweaks its parameters iteratively to minimize a given function to its local minimum.
* Overlearning – It is a term from machine learning/statistics that means that a learning system fits its learning data so well that it does not generalize to new data.
* Quality indicators - quality indicators refer to measurable metrics or criteria used to assess the performance, reliability, and suitability of a model for its intended purpose.
* Focal points - Focal points are central elements or areas of emphasis that draw attention and guide focus within a design, discussion, or analysis.
* Upscale the process - expanding and adapting the AI system to handle larger datasets or broader applications efficiently, ensuring consistent performance and scalability as the company grows.

**CONTEXT a**

The company is using AI to predict fair property prices based on available data. The AI expert has a strategy to model house prices, but there are concerns about accuracy and overfitting. The team will focus on preparing the data properly and then scaling the process.

**PROBLEM STATEMENT a**

How can the use of artificial intelligence be optimized to accurately predict real estate prices while avoiding the risks of overfitting and ensuring the reliability of regression models?

**CONSTRAINTS a**

* Dataset, all the parameters
* Quality indicators

**DELIVERABLES a**

* Make a jupyter notebook with the best fitting regression model.

**SOLUTION APPROACH a**

* Use linear regression
* Gradient descent

**ACTION PLAN a**

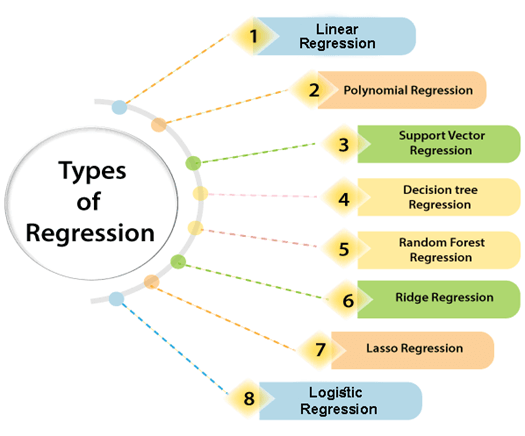
1. Study and compare different regression models.
2. Optimize each model.
3. Choose the best regression model and apply it to the dataset.

**STUDYING AND COMPARING REGRESSION MODELS A**

*What is a regression model?*

A regression model provides a function that describes the relationship between one or more independent variables and a response, dependent, or target variable. For example, the relationship between height and weight may be described by a linear regression model. A regression analysis is the basis for many types of prediction and for determining the effects on target variables.

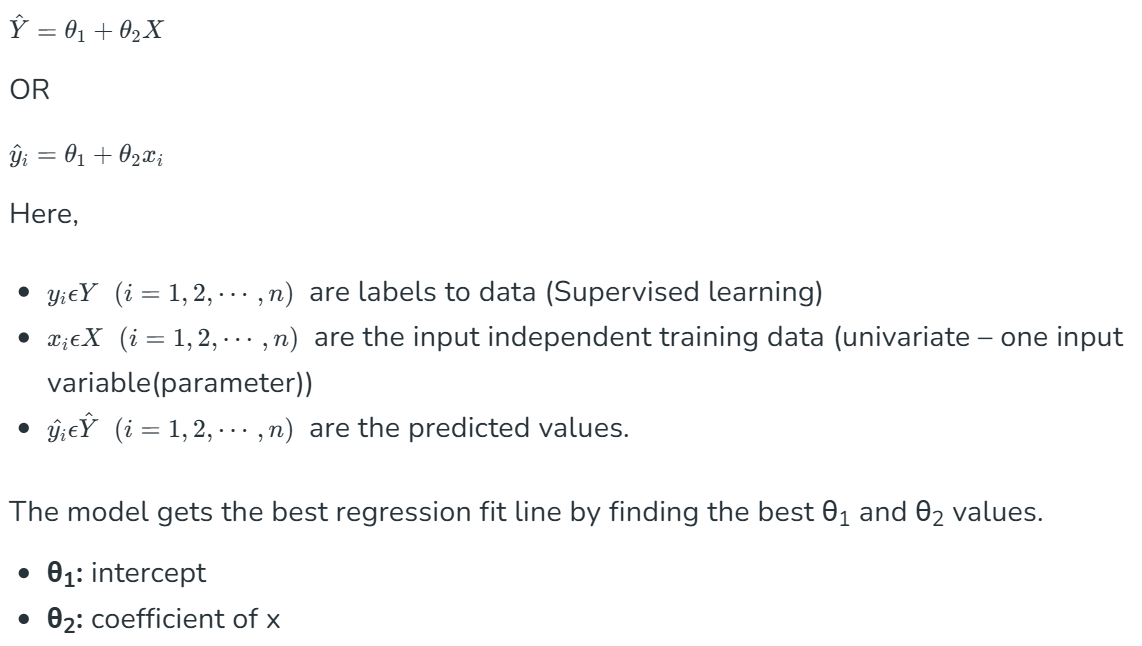
*Types of Regression Models*



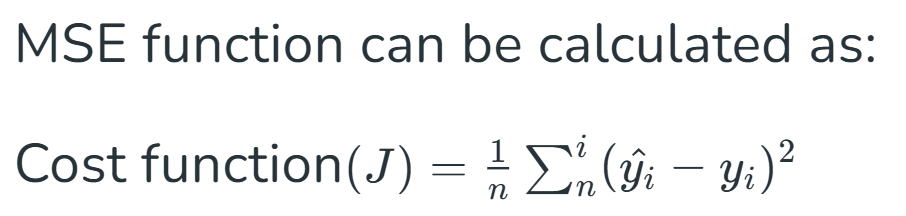
*Linear Regression*

* Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation to observed data.
* When there is only one independent feature, it is known as Simple Linear Regression, and when there is more than one feature, it is known as Multiple Linear Regression.
* Similarly, when there is only one dependent variable, it is considered Univariate Linear Regression, while when there are more than one dependent variables, it is known as Multivariate Regression.

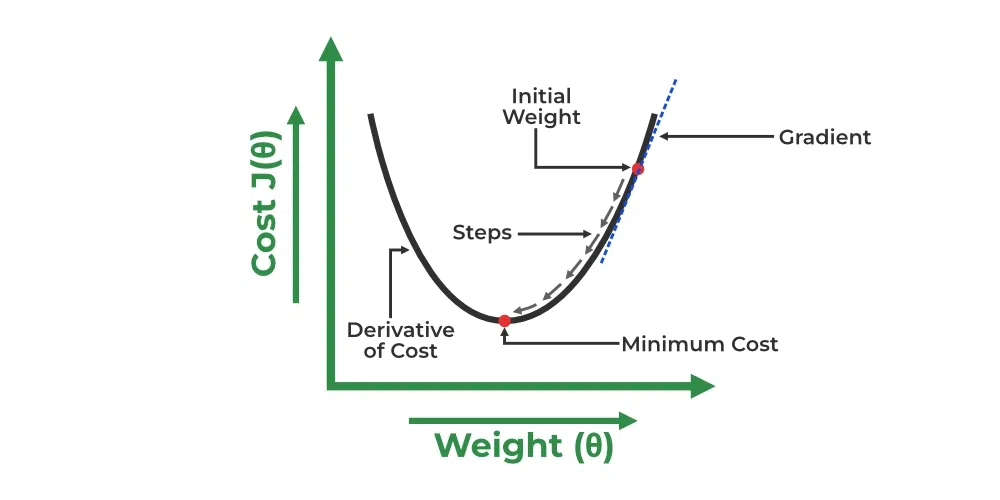
**Simple Linear Regression**



* The cost function or the loss function is nothing but the error or difference between the predicted value Ŷ and the true value Y.
* In Linear Regression, the Mean Squared Error (MSE) cost function is employed, which calculates the average of the squared errors between the predicted values and the actual values.
* The purpose is to determine the optimal values for the intercept θ1 and the coefficient of the input feature θ2 providing the best-fit line for the given data points.



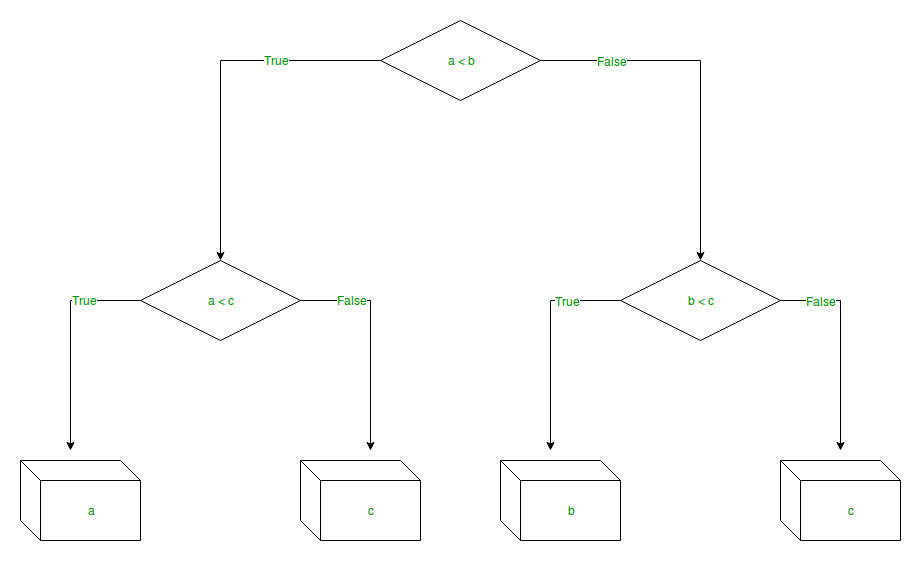
**Gradient Descent for Linear Regression**

A gradient is nothing but a derivative that defines the effects on outputs of the function with a little bit of variation in inputs. 

[Link](https://www.geeksforgeeks.org/ml-linear-regression/) for math

*Decision Tree Regression*

* Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility.
* Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.
* The branches/edges represent the result of the node and the nodes have either:
  + Conditions [Decision Nodes]
  + Result [End Nodes]

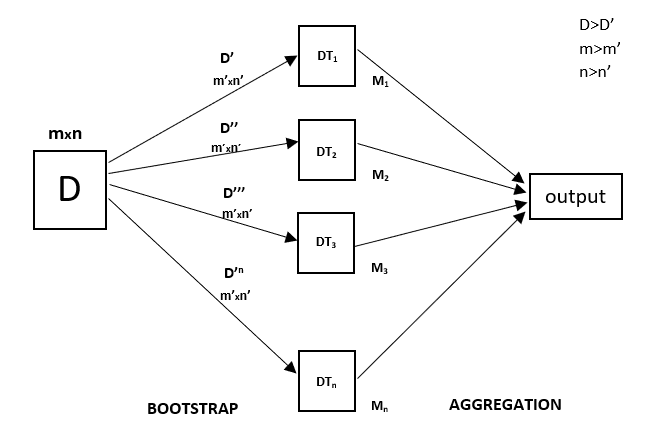


Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output.

* Discrete output example: A weather prediction model that predicts whether or not there’ll be rain on a particular day.
* Continuous output example: A profit prediction model that states the probable profit that can be generated from the sale of a product.

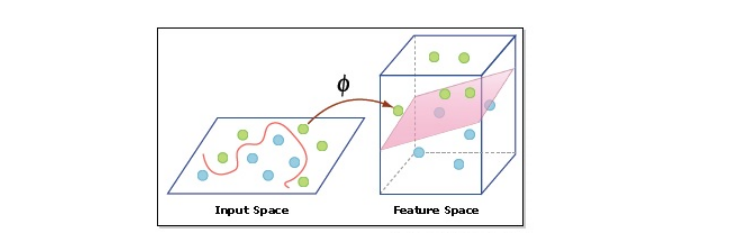
*Random Forest Regression*

* A random forest is an ensemble learning method that combines the predictions from multiple decision trees to produce a more accurate and stable prediction.
* It is a type of supervised learning algorithm that can be used for both classification and regression tasks.
* Every decision tree has high variance, but when we combine all of them in parallel then the resultant variance is low as each decision tree gets perfectly trained on that sample data, and hence the output does not depend on one decision tree but on multiple decision trees.
* We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.
* In the case of a regression problem, the final output is the mean of all the outputs. This part is called Aggregation.

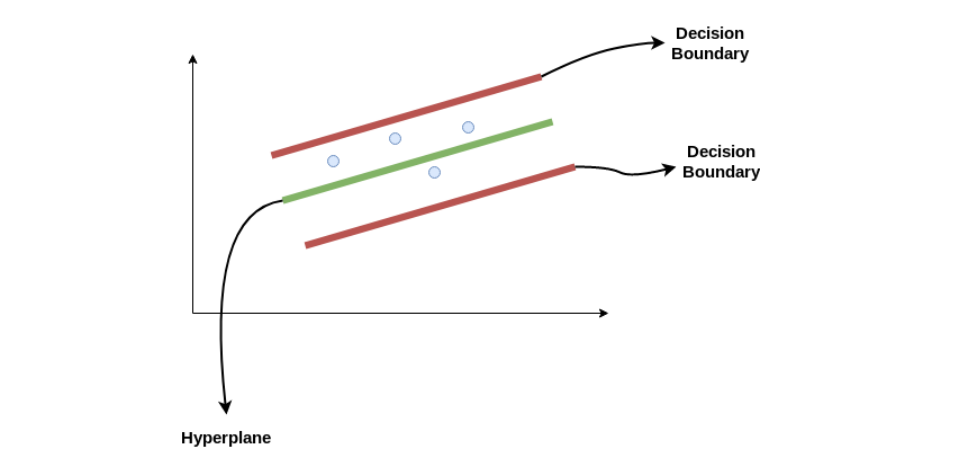


*Support Vector Regression*

* Support vector regression (SVR) is a type of support vector machine (SVM) that is used for regression tasks. It tries to find a function that best predicts the continuous output value for a given input value.
* SVR can use both linear and non-linear kernels.
  + A linear kernel is a simple dot product between two input vectors
  + a non-linear kernel is a more complex function that can capture more intricate patterns in the data.
* SVM works by finding a hyperplane in a high-dimensional space that best separates data into different classes.
* It aims to maximize the margin (the distance between the hyperplane and the nearest data points of each class) while minimizing classification errors.
* It’s widely used in tasks such as image classification, text categorization, etc.



* Unlike Support Vector Machines (SVMs) used for classification tasks, SVR Model seeks a hyperplane that best fits the data points in a continuous space.
* This is achieved by mapping the input variables to a high-dimensional feature space and finding the hyperplane that maximizes the margin (distance) between the hyperplane and the closest data points, while also minimizing the prediction error.



Consider these two red lines as the decision boundary and the green line as the hyperplane. When we move on with SVR in Machine Learning, our objective is to consider the points within the decision boundary line. Our best fit line is the hyperplane with the maximum number of points.

Comparison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Linear Regression** | **Decision Tree Regressor** | **Random Forest Regressor** | **Support Vector Regressor (Linear Kernel)** |
| **Model Type** | Parametric (linear model) | Non-parametric (tree-based) | Ensemble (multiple decision trees) | Non-parametric (linear hyperplane) |
| **Assumptions** | Assumes linear relationship between features and target | No assumptions about data distribution | No assumptions, based on multiple decision trees | Assumes linear relationship between features and target |
| **Interpretability** | High (simple, easy to understand) | Medium (easy to visualize, but can be complex) | Low (complex, black-box model) | Medium (easy for small datasets, harder for large) |
| **Handling non-linearity** | Poor (only works for linear relationships) | Good (handles non-linear relationships well) | Good (handles non-linear relationships through multiple trees) | Moderate (can capture some non-linearity but works best for linear) |
| **Overfitting Risk** | Low (but can underfit) | High (especially without pruning) | Medium (reduced by averaging trees) | Low (due to regularization) |
| **Performance on Large Datasets** | Fast (computationally efficient) | Slow (can become inefficient with large datasets) | Medium (computationally expensive) | Medium (requires careful parameter tuning) |
| **Handling Multicollinearity** | Poor (sensitive to highly correlated features) | Not affected by multicollinearity | Handles multicollinearity well | Not affected by multicollinearity |
| **Feature Scaling Required** | Yes (features must be scaled) | No (handles different feature scales) | No (handles different feature scales) | Yes (requires scaling for optimal performance) |
| **Tuning Complexity** | Low (few hyperparameters) | High (many parameters to tune) | Medium (several hyperparameters to tune) | Medium (needs tuning for kernel and margin) |