* What is the difference between R² and adjusted R²?
* What is overfitting in decision tree regression, and how to control it?
* What happens when α = 0 in Ridge Regression?

# Regression in machine learning

Regression in machine learning refers to a [**supervised learning**](https://www.geeksforgeeks.org/machine-learning/supervised-machine-learning/)technique where the goal is to predict a continuous numerical value based on one or more independent features. It finds relationships between variables so that predictions can be made. we have two types of variables present in regression:

* **Dependent Variable (Target)**: The variable we are trying to predict e.g house price.
* **Independent Variables (Features)**: The input variables that influence the prediction e.g locality, number of rooms.

Regression analysis problem works with if output variable is a real or continuous value such as “salary” or “weight”. Many different regression models can be used but the simplest model in them is linear regression.

**Types of Regression**

Regression can be classified into different types based on the number of predictor variables and the nature of the relationship between variables:

**1. Simple Linear Regression**

[**Linear regression**](https://www.geeksforgeeks.org/machine-learning/ml-linear-regression/)is one of the simplest and most widely used statistical models. This assumes that there is a linear relationship between the independent and dependent variables. This means that the change in the dependent variable is proportional to the change in the independent variables.For example predicting the price of a house based on its size.

**2. Multiple Linear Regression**

[Multiple linear regression](https://www.geeksforgeeks.org/machine-learning/ml-multiple-linear-regression-using-python/) extends simple linear regression by using multiple independent variables to predict target variable.For example predicting the price of a house based on multiple features such as size, location, number of rooms, etc.

**3. Polynomial Regression**

[Polynomial regression](https://www.geeksforgeeks.org/machine-learning/python-implementation-of-polynomial-regression/)is used to model with non-linear relationships between the dependent variable and the independent variables. It adds polynomial terms to the linear regression model to capture more complex relationships.For example when we want to predict a non-linear trend like population growth over time we use polynomial regression.

**4. Ridge & Lasso Regression**

[Ridge & lasso regression](https://www.geeksforgeeks.org/machine-learning/ridge-regression-vs-lasso-regression/) are regularized versions of linear regression that help avoid overfitting by penalizing large coefficients.When there’s a risk of overfitting due to too many features we use these type of regression algorithms.

**5. Support Vector Regression (SVR)**

SVR is a type of regression algorithm that is based on the [Support Vector Machine (SVM)](https://www.geeksforgeeks.org/machine-learning/support-vector-machine-algorithm/) algorithm. SVM is a type of algorithm that is used for classification tasks but it can also be used for regression tasks. SVR works by finding a hyperplane that minimizes the sum of the squared residuals between the predicted and actual values.

**6. Decision Tree Regression**

[Decision tree](https://www.geeksforgeeks.org/machine-learning/decision-tree/) Uses a tree-like structure to make decisions where each branch of tree represents a decision and leaves represent outcomes. For example predicting customer behavior based on features like age, income, etc there we use decison tree regression.

**7. Random Forest Regression**

[Random Forest](https://www.geeksforgeeks.org/machine-learning/random-forest-algorithm-in-machine-learning/) is a ensemble method that builds multiple decision trees and each tree is trained on a different subset of the training data. The final prediction is made by averaging the predictions of all of the trees. For example customer churn or sales data using this.

**Regression Evaluation Metrics**

Evaluation in machine learning measures the performance of a model. Here are some popular evaluation metrics for regression:

* [**Mean Absolute Error (MAE):**](https://www.geeksforgeeks.org/python/how-to-calculate-mean-absolute-error-in-python/) The average absolute difference between the predicted and actual values of the target variable.
* [**Mean Squared Error (MSE):**](https://www.geeksforgeeks.org/python/python-mean-squared-error/)The average squared difference between the predicted and actual values of the target variable.
* [**Root Mean Squared Error (RMSE)**](https://www.geeksforgeeks.org/software-engineering/rmse-root-mean-square-error-in-matlab/)**:** Square root of the mean squared error.
* [**Huber Loss:**](https://www.geeksforgeeks.org/machine-learning/sklearn-different-loss-functions-in-sgd/) A hybrid loss function that transitions from MAE to MSE for larger errors, providing balance between robustness and MSE’s sensitivity to outliers.
* [R2 – Score](https://www.geeksforgeeks.org/machine-learning/python-coefficient-of-determination-r2-score/): Higher values indicate better fit ranging from 0 to 1.

**Regression Model Machine Learning**

Let's take an example of linear regression. We have a [**Housing data set**](https://vincentarelbundock.github.io/Rdatasets/csv/Ecdat/Housing.csv) and we want to predict the price of the house. Following is the python code for it.



import matplotlib

matplotlib.use('TkAgg') # General backend for plots

import matplotlib.pyplot as plt

import numpy as np

from sklearn import datasets, linear\_model

import pandas as pd

# Load dataset

df = pd.read\_csv("Housing.csv")

​

# Extract features and target variable

Y = df['price']

X = df['lotsize']

​

# Reshape for compatibility with scikit-learn

X = X.to\_numpy().reshape(len(X), 1)

Y = Y.to\_numpy().reshape(len(Y), 1)

​

# Split data into training and testing sets

X\_train = X[:-250]

X\_test = X[-250:]

Y\_train = Y[:-250]

Y\_test = Y[-250:]

​

# Plot the test data

plt.scatter(X\_test, Y\_test, color='black')

plt.title('Test Data')

plt.xlabel('Size')

plt.ylabel('Price')

plt.xticks(())

plt.yticks(())

​

# Train linear regression model

regr = linear\_model.LinearRegression()

regr.fit(X\_train, Y\_train)

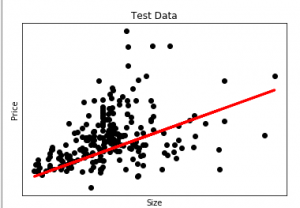
​

# Plot predictions

plt.plot(X\_test, regr.predict(X\_test), color='red', linewidth=3)

plt.show()

**Output:**



Here in this graph we plot the test data. The red line indicates the best fit line for predicting the price.

To make an individual prediction using the linear regression model:

**print("Predicted price for a lot size of 5000: " + str(round(regr.predict([[5000]])[0][0])))**

**Applications of Regression**

* **Predicting prices:** Used to predict the price of a house based on its size, location and other features.
* **Forecasting trends:** Model to forecast the sales of a product based on historical sales data.
* **Identifying risk factors:** Used to identify risk factors for heart patient based on patient medical data.
* **Making decisions:** It could be used to recommend which stock to buy based on market data.

**Advantages of Regression**

* Easy to understand and interpret.
* Robust to outliers.
* Can handle both linear relationships easily.

**Disadvantages of Regression**

* Assumes linearity.
* Sensitive to situation where two or more independent variables are highly correlated with each other i.e multicollinearity.
* May not be suitable for highly complex relationships.

**Conclusion**

Regression in machine learning is a fundamental technique for predicting continuous outcomes based on input features. It is used in many real-world applications like price prediction, trend analysis and risk assessment. With its simplicity and effectiveness regression is used to understand relationships in data.

[geekforgeeksregre](https://www.geeksforgeeks.org/machine-learning/regression-in-machine-learning/)

[builtinregre](https://builtin.com/data-science/regression-machine-learning)