MACHINE LEARNING - LINEAR REGRESSION

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog). There are two main types:

Simple regression

Simple linear regression uses traditional slope-intercept form, where m and b are the variables our algorithm will try to "learn" to produce the most accurate predictions. x represents our input data and y represents our prediction.

$Y = mx + b$

Multivariable regression

A more complex, multi-variable linear equation might look like this, where w represents the coefficients, or weights, our model will try to learn.

$f(x,y,z) = w_1x + w_2y + w_3z$

PROJECT : THE BOSTON HOUSING DATASET

The Boston housing dataset contains the information about different houses in Boston based on crime rate, tax, number of rooms etc.

I have used this dataset to predict the housing prices of Boston based on 13 such attributes.

Algorithms used for predicting the parameters :

-> Gradient descent
-> Normal equation formula

Programming language used :

-> Octave

Gradient descent :

Gradient descent is an optimization algorithm which is commonly-used to train machine learning models and neural networks. Training data helps these models learn over time, and the cost function within gradient descent specifically acts as a barometer, gauging its accuracy with each iteration of parameter updates. Until the function is close to or equal to zero, the model will continue to adjust its parameters to yield the smallest possible error.

Normal equation :

Normal Equation is an analytical approach to Linear Regression with a Least Square Cost Function. We can directly find out the value of θ without using Gradient Descent.

Normal Equation is a follows :

$$\theta = \left(X^T X\right)^{-1} \cdot \left(X^T y\right)$$

In the above equation,
θ : hypothesis parameters that define it the best.
X : Input feature value of each instance.
Y : Output value of each instance.

Output obtained from Gradient descent approach :

Minimized cost function J = 11.018

The parameters (theta vector):

3.6460e+01 = 36.460
-1.0843e-01 = -0.10843
4.9262e-02 = 0.049262
2.0237e-02 = 0.020237
2.6000e+00 = 2.6000
-1.7767e+01 = -17.767
3.6008e+00 = 3.6008
2.7053e-03 = 0.0027053
-1.4992e+00 = -1.4992
2.9943e-01 = 0.29943
-1.0510e-02 = -0.010510
-9.9813e-01 = -0.99813
1.3080e-02 = 0.013080
-5.2467e-01 = -0.52467

Hypothesis equation :

$$Y = 36.460 - 0.10843*x_1 + 0.049262*x_2 + 0.020237*x_3 + 2.6000*x_4 - 17.767*x_5 + 3.6008*x_6 + 0.0027053*x_7 - 1.4992*x_8 + 0.29943*x_9 - 0.010510*x_{10} - 0.99813*x_{11} + 0.013080*x_{12} - 0.52467*x_{13}$$

Output obtained from normal equation approach :

Minimized cost function J = 10.947

The parameters (theta vector) :

3.6459e+01 = 36.459
-1.0801e-01 = -0.10801
4.6420e-02 = 0.046420
2.0559e-02 = 0.020559
2.6867e+00 = 2.6867
-1.7767e+01 = -17.767
3.8099e+00 = 3.8099
6.9222e-04 = 0.00069222
-1.4756e+00 = -1.4756
3.0605e-01 = 0.30605
-1.2335e-02 = -0.0123345
-9.5275e-01 = -0.95275
9.3117e-03 = 0.0093117
-5.2476e-01 = -0.52476

Hypothesis equation :

$Y = 36.459 - 0.10801*x_1 + 0.046420*x_2 + 0.020559*x_3 + 2.6867*x_4 - 17.767*x_5 + 3.8099*x_6 + 0.00069222*x_7 - 1.4756*x_8 + 0.30605*x_9 - 0.0123345*x_{10} - 0.95275*x_{11} + 0.0093117*x_{12} - 0.52476*x_{13}$

Done by :

PA SURAAJ VIKAS RAJU

SASTRA deemed university
Btech Computer science and engineering 2^{nd} year