Learn To Make Prediction By Using Multiple Variables

**Introduction:**

The goal of the blogpost is to equip beginners with basics of Linear Regression algorithm having multiple features and quickly help them to build their first model. This is also known as multivariable Linear Regression. We will mainly focus on the modeling side of it . The data cleaning and preprocessing parts would be covered in detail in an upcoming post.

A multivariable model can be thought of as a model in which multiple variables are found on the right side of the model equation. This type of statistical model can be used to attempt to assess the relationship between a number of variables. A simple linear regression model has a continuous outcome and one predictor, whereas a multiple or multivariable linear regression model has a continuous outcome and multiple predictors (continuous or categorical). A simple linear regression model would have the form



a multivariable or multiple linear regression model would take the form



where y is a continuous dependent variable, x is a single predictor in the simple regression model, and x1, x2, …, xk are the predictors in the multivariable model.

In statistics, the mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and what is estimated.

Multivariable linear regression can can model more complex relationship which comes from various features together. They should be used in cases where one particular variable is not evident enough to map the relationship between the independent and the dependent variable.

Let’s work on a case study to understand this better.

**Problem Statement:**

To predict the relative performance of a computer hardware given other associated attributes of the hardware.

**Data details**

Computer Hardware dataset  
===========================  
  
URL : <https://archive.ics.uci.edu/ml/datasets/Computer+Hardware>

1. Title: Relative CPU Performance Data   
  
2. Source Information  
 -- Creators: Phillip Ein-Dor and Jacob Feldmesser  
 -- Ein-Dor: Faculty of Management; Tel Aviv University; Ramat-Aviv;   
 Tel Aviv, 69978; Israel  
 -- Donor: David W. Aha (aha@ics.uci.edu) (714) 856-8779   
 -- Date: October, 1987  
  
3. Past Usage:  
 1. Ein-Dor and Feldmesser (CACM 4/87, pp 308-317)  
 -- Results:   
 -- linear regression prediction of relative cpu performance  
 -- Recorded 34% average deviation from actual values   
 2. Kibler,D. & Aha,D. (1988). Instance-Based Prediction of  
 Real-Valued Attributes. In Proceedings of the CSCSI (Canadian  
 AI) Conference.  
 -- Results:  
 -- instance-based prediction of relative cpu performance  
 -- similar results; no transformations required  
 - Predicted attribute: cpu relative performance (numeric)  
  
4. Relevant Information:  
 -- The estimated relative performance values were estimated by the authors  
 using a linear regression method. See their article (pp 308-313) for  
 more details on how the relative performance values were set.  
  
5. Number of Instances: 209   
  
6. Number of Attributes: 10 (6 predictive attributes, 2 non-predictive,   
 1 goal field, and the linear regression guess)  
  
7. Attribute Information:  
 1. vendor name: 30   
 (adviser, amdahl,apollo, basf, bti, burroughs, c.r.d, cambex, cdc, dec,   
 dg, formation, four-phase, gould, honeywell, hp, ibm, ipl, magnuson,   
 microdata, nas, ncr, nixdorf, perkin-elmer, prime, siemens, sperry,   
 sratus, wang)  
 2. Model Name: many unique symbols  
 3. MYCT: machine cycle time in nanoseconds (integer)  
 4. MMIN: minimum main memory in kilobytes (integer)  
 5. MMAX: maximum main memory in kilobytes (integer)  
 6. CACH: cache memory in kilobytes (integer)  
 7. CHMIN: minimum channels in units (integer)  
 8. CHMAX: maximum channels in units (integer)  
 9. PRP: published relative performance (integer)  
 10. ERP: estimated relative performance from the original article (integer)  
  
8. Missing Attribute Values: None  
  
9. Class Distribution: the class value (PRP) is continuously valued.  
 PRP Value Range: Number of Instances in Range:  
 0-20 31  
 21-100 121  
 101-200 27  
 201-300 13  
 301-400 7  
 401-500 4  
 501-600 2  
 above 600 4  
  
Summary Statistics:  
 Min Max Mean SD PRP Correlation  
 MCYT: 17 1500 203.8 260.3 -0.3071  
 MMIN: 64 32000 2868.0 3878.7 0.7949  
 MMAX: 64 64000 11796.1 11726.6 0.8630  
 CACH: 0 256 25.2 40.6 0.6626  
 CHMIN: 0 52 4.7 6.8 0.6089  
 CHMAX: 0 176 18.2 26.0 0.6052  
 PRP: 6 1150 105.6 160.8 1.0000  
 ERP: 15 1238 99.3 154.8 0.9665

**Tools used** :

Pandas , Numpy , Matplotlib , scikit-learn

**Python Implementation with code :**

1. **Import necessary libraries**

Import the necessary modules from specific libraries.

|  |
| --- |
|  |

1. **Load the data set**

Use pandas module to read the taxi data from the file system. Check few records of the dataset.

|  |
| --- |
|  |

1. **Feature Selection**

Select only the numerical fields for model fitting.

|  |
| --- |
|  |

1. **Select the predictor and target variables**

|  |
| --- |
|  |

1. **Train test split :**

|  |
| --- |
|  |

**6. Normalize the data :**

Before we do the fitting, normalize the data so that the data is centered around the mean and has unit standard deviation.

|  |
| --- |
|  |

**7. Training / model fitting :**

Fit the model to selected supervised data

|  |
| --- |
|  |

**8. Model parameters study :**

The coefficient R^2 is defined as (1 - u/v), where u is the residual sum of squares ((y\_true - y\_pred) \*\* 2).sum() and v is the total sum of squares ((y\_true - y\_true.mean()) \*\* 2).sum().

|  |
| --- |
|  |

**9. Accuracy report with test data:**

Visualize the goodness of the fit with the predictions being visualized by a line

|  |
| --- |
|  |

**10. What are your Conclusion**s?