Linear and Non-Linear Regression Models

 Lisa Hall

University of Cumberlands

MSDS-530-A01 - First Bi-term

Dr. Afshin Zarenejad

February 19, 2025

**Linear and Non-Linear Regression Models**

Understanding linear and non-linear help data workers understand relationships between independent and dependent variables. Linear regression illustrates a consistent rate of change, like how much y will change when x changes by a certain amount. Non-linear regression helps illustrate more complicated relationships between variables; there are situations where x and y relate nonlinearly (Fávero & Belfiore, 2018). This paper will describe regression models and analyze their applications in real-world scenarios.

**Linear Regression**

Two primary types of linear regression models exist: simple linear regression and multiple linear regression. Simple linear regression models describe the relationship between two quantitative variables (Penn State, 2018). Multiple linear regression models have similar assumptions but use two or more predictors. The regression equation for simple linear regression is seen in Equation 1. The equation for multiple linear regression can be seen in Equation 2.

Y = β0 + β1X + ϵ [1]

Y = β0 + β1X + β2X2 + … + βnXn + ϵ[2]

We can interpret Equation 1 as Y equals β1 times X plus β0. β0 is the intercept, β1 the slope. X is an independent variable, y is the dependent variable's predicted value, and ϵ is the difference between the actual and predicted values. Equation 2 shows how multiple linear regression extends the equation to include multiple independent variables (Bruce & Bruce, 2019).

**Nonlinear Regression**

Nonlinear regression models are helpful when relationships between variables are not linear; often, their relationships are not best described as a straight line. These models can take more complex data and illustrate them in ways that allow for more intuitive interpretation. There are several types of nonlinear models: quadratic, exponential, logarithmic, polynomial, and logistic. Equation 3 shows the nonlinear regression equation: X is a vector of p predictors, β is a vector of k parameters, f() is some known regression function, and ϵ represents the difference between actual and predicted values.

Y = f(X, β) + ϵ [3]

**Model Estimation**

To best understand these models and their applications, let us review an example of a multiple linear regression model. To exemplify this, we will use the sklearn dataset diabetes (Scikit-learn developers, n.d.). We will also utilize the sklearn linear model library (Scikit-learn developers, n.d.). We can view the import of the dataset and libraries in Appendix A and will reference the Python code in Appendix A for the model interpretation. Next, we set the x independent variables, which are features like age, BMI, and blood pressure, and then store the dependent variables using y, representing the diabetes progression score. We can view this as the x and y input of the regression model.

Next, we will select three features for interpretation and set them as a subset, using age, BMI, and BP. We can then train the linear regression model, setting a variable as a model and using the linearregression() function, this creates an instance of the linear regression model. Model.fit(x\_subset, y) trains the model by finding the best-fit line and calculating the intercept and coefficients. The model interprets how age, BMI, and blood pressure relate to diabetes progression. We can observe in Appendix A the intercept variable, which is set as the model intercept. The coefficients variable uses the x\_subset columns and the model.coef\_ to target changes to corresponding features. Then, the model score will return the r squared value, which is a numeric between 0 and 1; higher values indicate a better fit. Then, we can observe the print for intercept, coefficients, and r\_sqared and interpret what they mean.

Now, we can answer the question of what the intercept, r\_squared, and coefficients are and what they mean. The intercept is 152.13, representing the baseline for diabetes progression when all predictors are zero. The coefficients are Age 25.99, BMI 788.78, and BP 394.13; we can interpret that for each unit of increase, the diabetes progression increases by the coefficient. BMI appears to have the most significant impact on progression. R squared is rounded to .40, which means that there could be other factors influencing diabetes progression.

Consider how these modeling methods could influence healthcare. There has been incredible benefit in understanding the influence of disease progression. This would allow scientists in the medical field to focus their efforts on addressing influencing factors that could help establish best practices and improve patient care. For example, in this case, it could be beneficial to address BMI and blood pressure, helping diabetics reduce the severity of their disease and potentially improve their quality of life.

**Conclusion**

Utilizing linear and nonlinear regression modes can contextualize the relationships between dependent and independent variables. Using tools like Python, data workers can access libraries that enable them to produce insights quickly and efficiently. These models have a wide range of applications, especially in healthcare where understanding contributing factors in disease can help save lives. Understanding the outputs of these models can help address the significance of the relationships between variables.

**References**

Bruce, P., & Bruce, A. (2019). *Practical statistics for data scientists: 50+ essential concepts using R and Python* (2nd ed.). O'Reilly Media.

Fávero, L. P., & Belfiore, P. (2018). *Data Science for Business and Decision Making* (1st ed.). Academic Press. https://doi.org/10.1016/C2016-0-01101-4

Penn State. (2018). *Lesson 2: Simple linear regression (SLR) model*. STAT 462: Applied Regression Analysis. https://online.stat.psu.edu/stat462/node/79/

Scikit-learn developers. (n.d.). *Diabetes dataset*. Scikit-learn. https://scikit-learn.org/stable/datasets/toy\_dataset.html#diabetes-dataset

Scikit-learn developers. (n.d.). *sklearn.linear\_model: Linear models* (Version 1.3.0) [Python library]. Scikit-learn. https://scikit-learn.org/stable/modules/linear\_model.html

**Appendix A**

**Python Code for Diabetes Linear Regression**

**import pandas as pd**

**from sklearn.linear\_model import LinearRegression**

**from sklearn.datasets import load\_diabetes**

**diabetes = load\_diabetes()**

**X = pd.DataFrame(diabetes.data, columns=diabetes.feature\_names)**

**y = pd.Series(diabetes.target)**

**X\_subset = X[['age', 'bmi', 'bp']]**

**model = LinearRegression()**

**model.fit(X\_subset, y)**

**intercept = model.intercept\_**

**coefficients = dict(zip(X\_subset.columns, model.coef\_))**

**r\_squared = model.score(X\_subset, y)**

**intercept, coefficients, r\_squared**

**print(intercept, coefficients, r\_squared)**