Andhra Pradesh State Skill Development Corporation

Skill AP
APSSDC

eSkill AP
Learn Anytime Anywhere
Andhra Pradesh State Skill Development Corporation

Machine Learning

Supervised Learning: Regression

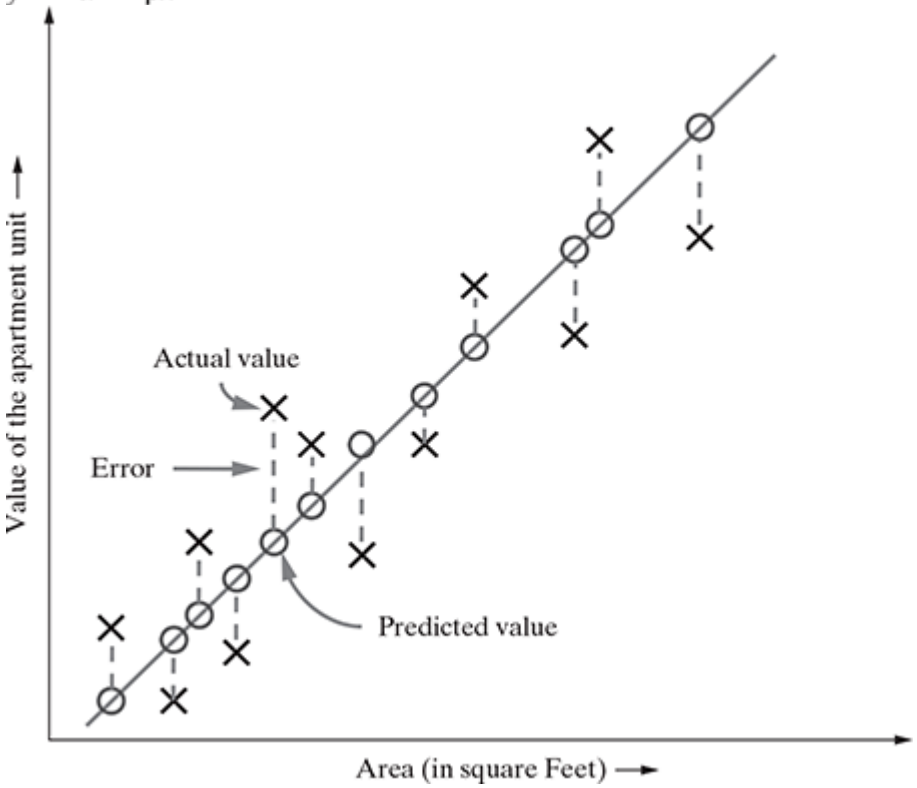
CHAPTER 5

Supervised Learning: Regression

Introduction to Regression:

A well-fitted regression model churns out predicted values close to actual values. Hence, a regression model which ensures that the difference between predicted and actual values is low can be considered as a good model. Graph represents a very simple problem of real estate value prediction solved using a linear regression model. If 'area' is the predictor variable (say x) and 'value' is the target variable (say y), the linear regression model can be represented in the form:

$$y = \alpha + \beta x$$



For a certain value of x, say $\hat{x}$, the value of y is predicted as $\hat{y}$ whereas the actual value of y is Y (say). The distance between the actual value and the fitted or predicted value, i.e. $\hat{y}$ is known as residual. The regression model can be considered to be fitted well if the difference between actual and predicted value, i.e. the residual value is less.

R-squared is a good measure to evaluate the model fitness. It is also known as the coefficient of determination, or for multiple regression, the coefficient of multiple determination.

The R-squared value lies between 0 to 1 (0%–100%) with a larger value representing a better fit.

It is calculated as: $$R^2 = \frac{SST - SSE}{SST}$$

Sum of Squares Total (SST) = squared differences of each observation from the overall mean = $\sum_{i=1}^{n} (y_i - \bar{y})^2$ where $\bar{y}$ is the mean.

Sum of Squared Errors (SSE) (of prediction) = sum of the squared residuals = $\sum_{i=1}^{n} (Y_i - \hat{y})^2$ where is the predicted value of yi and Yi is the actual value of yi.

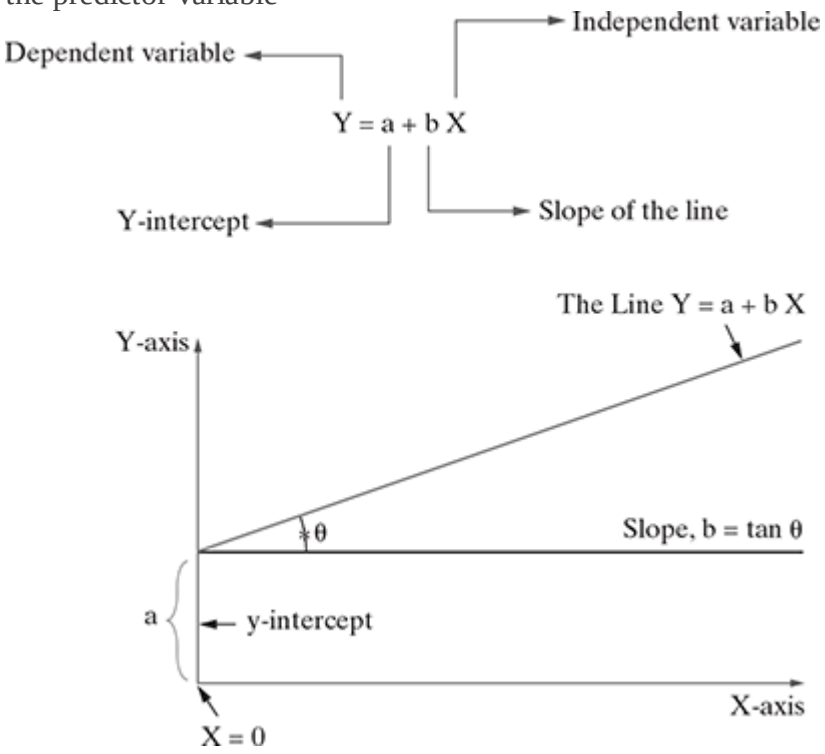
The most common regression algorithms are

- Simple linear regression
- Multiple linear regression
- Polynomial regression
- Multivariate adaptive regression splines
- Logistic regression
- Maximum likelihood estimation (least squares)

Understanding Linear Regression and Linear Regression Implementation

Linear Regression

As the name indicates, simple linear regression is the simplest regression model which involves only one predictor. This model assumes a linear relationship between the dependent variable and the predictor variable



In the context of Karen's problem, if we take Price of a Property as the dependent variable and the Area of the Property (in sq. m.) as the predictor variable, we can build a model using simple linear regression.

PriceProperty = *f*(AreaProperty)

Assuming a linear association, we can reformulate the model as

PriceProperty = *a* + *b*. AreaProperty

where '*a*' and '*b*' are intercept and slope of the straight line, respectively.

Just to recall, straight lines can be defined in a slope–intercept form $Y = (a + bX)$, where a = intercept and b = slope of the straight line. The value of intercept indicates the value of Y when $X = 0$. It is known as 'the intercept or Y intercept' because it specifies where the straight line crosses the vertical or Y-axis

Introduction to Polynomials :

Polynomial regression is a special case of linear regression where we fit a polynomial equation on the data with a curvilinear relationship between the target variable and the independent variables.

In a curvilinear relationship, the value of the target variable changes in a non-uniform manner with respect to the predictor (s).

In Linear Regression, with a single predictor, we have the following equation:

$$Y = \theta_0 + \theta_1 x$$

where,

Y is the target,

x is the predictor,

□0 is the bias,

and □1 is the weight in the regression equation

This linear equation can be used to represent a linear relationship. But, in polynomial regression, we have a polynomial equation of degree n represented as:

$$Y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_n x^n$$

Here:

□0 is the bias,

□1, □2, …, □n are the weights in the equation of the polynomial regression,

and n is the degree of the polynomial

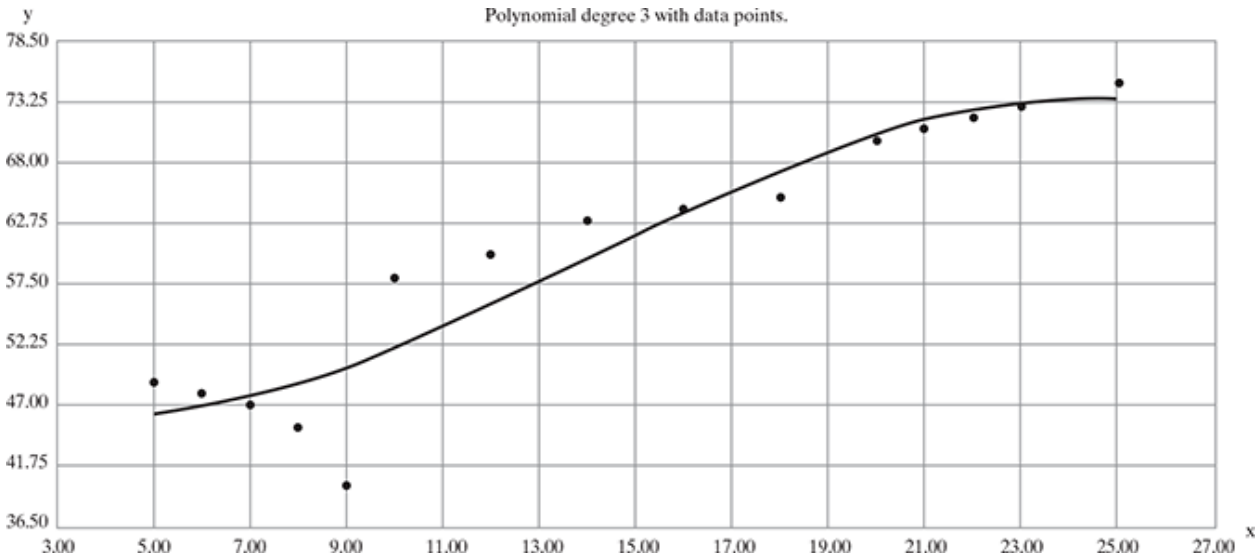
The number of higher-order terms increases with the increasing value of n, and hence the equation becomes more complicated.

Simple example for Polynomial Regression

Example: Let us use the below data set of (X, Y) for degree 3 polynomial.

Internal Exam (*X*)	15	23	18	23	24	22	22	19	19	16	24	11	24	16	23
External Exam (*Y*)	49	63	58	60	58	61	60	63	60	52	62	30	59	49	68

As you can observe, the regression line is slightly curved for polynomial degree 3 with the above 15 data points. The regression line will curve further if we increase the polynomial degree. At the extreme value as shown below, the regression line will be overfitting into all the original values of *X*.



Polynomial Regression - Data Collection:

There is a Human Resource company, which is going to hire a new candidate. The candidate has told his previous salary 160K per annum, and the HR have to check whether he is telling the truth or bluff. So to identify this, they only have a dataset of his previous company in which the salaries of the top 10 positions are mentioned with their levels. By checking the dataset available, we have found that there is a non-linear relationship between the Position levels and the salaries. Our goal is to build a Bluffing detector regression model, so HR can hire an honest candidate. Below are the steps to build such a model.

Position	Level(X-variable)	Salary(Y-Variable)
Business Analyst	1	45000
Junior Consultant	2	50000
Senior Consultant	3	60000
Manager	4	80000
Country Manager	5	110000
Region Manager	6	150000
Partner	7	200000
Senior Partner	8	300000
C-level	9	500000
CEO	10	1000000

Data Pre-processing and Correlation:

The data pre-processing step will remain the same as in previous regression models, except for some changes. In the Polynomial Regression model, we will not use feature scaling, and also we will not split our dataset into training and test sets. It has two reasons:

- The dataset contains very little information which is not suitable to divide it into a test and training set, else our model will not be able to find the correlations between the salaries and levels.
- In this model, we want very accurate predictions for salary, so the model should have enough information.

Correlation :

Polynomial regression of degree p in one independent variable χ is considered. Numerically large sample correlations between $\chi\alpha$ and $\chi\beta$, $\alpha < \beta$, a, $\beta = 1, \cdots, p$, may cause ill-conditioning in the matrix to be inverted in application of the method of least squares. These sample correlations are investigated. It is confirmed that centering of the independent variable to have zero sample mean removes nonessential ill-conditioning. If the sample values of χ are placed symmetrically about their mean, the sample correlation between $\chi\alpha$ and $\chi\beta$ is reduced to zero by centering when $\alpha + \beta$ is odd, but may remain large when $\alpha + \beta$ is even

Modelling and Evaluation of Performance of Models - Train Test Split

Step 1: Importing the libraries

In this first step, we will be importing the libraries required to build the ML model. The NumPy library and the matplotlib are imported. Additionally, we have imported the *Pandas* library for data analysis.

Step 2: Importing the dataset

In this step, we shall use pandas to store the data obtained from my github repository and store it as a Pandas DataFrame using the function "**pd.read_csv**".

Step 3: Training the Polynomial Regression model on the whole dataset

The dataset on which we are using has very few numbers of rows and hence we train the entire dataset for building the Polynomial Regression model. In this the "**PolynomialFeatures**" function is used to assign the degree of the polynomial line that we are going to plot. In this, the degree is set as ***4***.

The independent variable X is then fitted with the PolynomialFeatures class and is converted to a new variable **X_poly**. In this, the variable X is converted to a new matrix X_Poly which consists of all the polynomial combinations of features with degree=4.

Step 4: Predicting the Results

In this step, we are going to predict the values Salary based on the Polynomial Regression model built. The "**regressor.predict**" function is used to predict the values for our independent

variable, X_poly. We assign the predicted values as y_pred. We now have two data, y(real values) and y_pred (predicted values)

Step 6: Visualizing the Polynomial Regression results

In this last step, we shall visualize the polynomial model that was built using the given data and plot the values of "***y***" and "**y_pred**" on the graph and analyze the results