Linear Regression

Soumen Ghosh
Indian Institute of Information Technology, Sri City

I. Introduction

LINEAR regression is a linear approach for modeling the relationship between a scalar dependent variable y and one or more independent variables (or explanatory variables) denoted X. The Linear Regression modeling approach further divided into three types: (1) Simple Linear Regression, (ii) Multiple Linear Regression and (iii) Multivariate Linear Regression.

- A Simple Linear Regression is the most basic model of linear regression. It is just two variables and is modeled as a linear relationship with an error term:
$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \tag{1}$$
where x and y are the independent variable and dependent variable respectively. The aim is to fit the model, which will give us the best estimates for β_0 and β_1. The above equation can rewriten as:
$$y = f(x) \tag{2}$$
- The Multiple Linear Regression (aka multivariable regression), where we have one dependent variable and multiple independent variables:
$$y = \beta_0 + \beta_1 x_i^{(1)} + \beta_2 x_i^{(2)} + ... + \beta_n x_i^{(n)} + \varepsilon_i \tag{3}$$
which is also represent as:
$$y = f(x_1, x_2, ..., x_n) \tag{4}$$
- Multivariate Linear Regression is quite similar to the simple linear regression model, but with multiple independent variables contributing to the dependent variable and hence multiple coefficients to determine and complex computation due to the added variables. The equation of multivariate linear regression is.
$$Y_i = \beta_0 + \beta_1 x_i^{(1)} + \beta_2 x_i^{(2)} + ... + \beta_n x_i^{(n)} + \varepsilon_i \tag{5}$$
so the expression may be written as
$$Y = f(X), \tag{6}$$
where X and Y are the matrices of independent variable and dependent variable respectively.

The aim of this work is to implement the multivariate linear regression modeling approach for Forest Fires Data Set[2] dataset and analysis the obtained results.

II. Methodology

In this work, we have implemented multiple Polynomial regression for predict the burned area of forest. We have used Python as a programming framework for this implementation. Polynomial regression fits a nonlinear relationship between dependent variable and independent variables, and has been used to describe nonlinear phenomena.

$$y = \beta_0 + \beta_1 X^1 + \beta_2 X^2 + ... + \beta_n X^n + \varepsilon \tag{7}$$

We have implemented the above mentions formula and tested in two dataset for different degrees of polynomial. The source code and the obtained result has been reported in the Appendix-A. The analysis of the results is given in Figure-1 and 2.

III. Experiments, Results and Analysis

We have used two datasets for this experiments. Both the datasets are collected from the UCI Machine Learning Repository. The description of the datasets are given bellow:

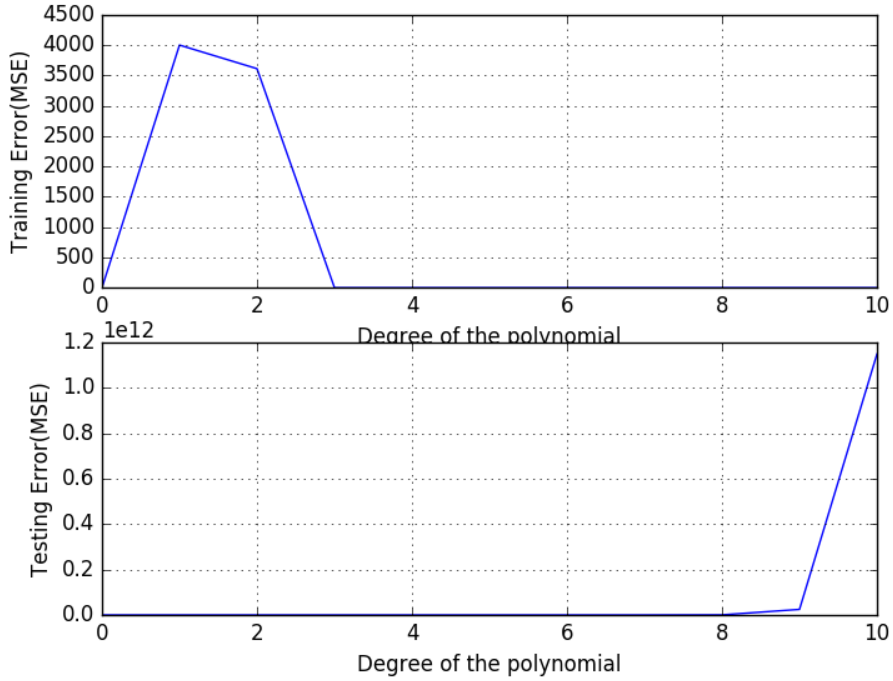

Fig. 1: MSE Error Vs Degree of Polynomial for Forest Fires Dataset

A. *Dataset Description*

1) Dataset-I: In this work, we have used Forest Fires Data Set[2]. The dataset have 13 attributes and 570 observations. This dataset is collected from UCI Machine Learning Repository. This is use for regression task, where the aim is to predict the burned area of forest fires, in the northeast region of Portugal, by using meteorological and other data. The attribute information of the dataset is given bellow:

- X - x-axis spatial coordinate within the Montesinho park map: 1 to 9
- Y - y-axis spatial coordinate within the Montesinho park map: 2 to 9
- Month - month of the year: 'jan' to 'dec'(1 to 12)
- Day - day of the week: 'mon' to 'sun' (1 to 7)
- FFMC - FFMC index from the FWI system: 18.7 to 96.20
- DMC - DMC index from the FWI system: 1.1 to 291.3
- DC - DC index from the FWI system: 7.9 to 860.6
- ISI - ISI index from the FWI system: 0.0 to 56.10
- Temp - temperature in Celsius degrees: 2.2 to 33.30
- RH - relative humidity in %: 15.0 to 100
- Wind - wind speed in km/h: 0.40 to 9.40
- Rain - outside rain in mm/m2 : 0.0 to 6.4
- Area - the burned area of the forest (in ha): 0.00 to 1090.84 (this output variable is very skewed towards 0.0, thus it may make sense to model with the logarithm transform).

2) Dataset-II: In this project, we have taken one more dataset from UCI Machine Learning Repository. The name of the dataset is "Daily Demand Forecasting Orders Data Set". The dataset having 60 observations and 13 attributes. This is a time series data. The dataset was collected during 60 days, this is a real database of a brazilian logistics company. The dataset has twelve predictive attributes and a target that is the total of orders for daily treatment.

B. *Results and Analysis*

In this project, we used two datasets for experiment. We have divided both the dataset in the ratio of 70:30 for training set and testing set respectively. The polynomial regression model was built for the range of degree one to degree ten and computed the MSE error for each case. The obtained results was plotted in the Figure-1 and 2. It was observed from the results that for high degree of polynomial the error also increasing.

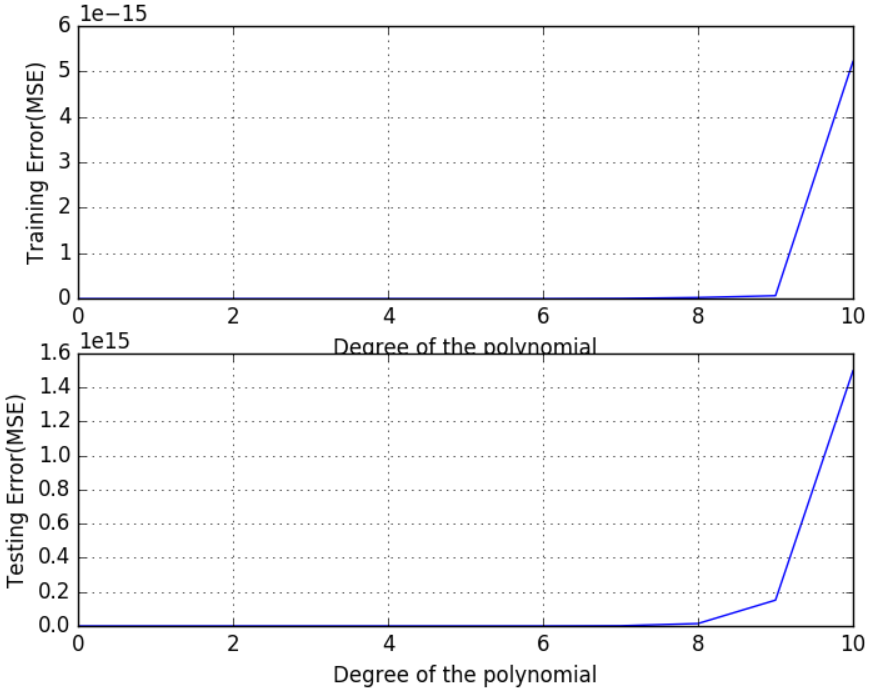

Fig. 2: MSE Error Vs Degree of Polynomial for Daily Demand Forecasting Orders

IV. Conclusion

It was observed from this experiment that the very high degree of polynomial cases overfitting. This polynomial regression model is performing decent for this two datasets upto degree eight.

Appendix A
Python Code

```
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
from sklearn import linear_model
from random import seed
from random import randrange
from csv import reader
from math import sqrt
from sklearn import linear_model
import pandas as pd
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt


data = pd.read_csv('test.csv', header = None)
print("Shape of the training data is {}".format(data.shape))
[m ,n] = data.shape
X = data.iloc[:, 0:n-1].values
y = data.iloc[:, n-1:n].values

from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
                                                    random_state=1)

test_list = [0]
```

```
train_list = [0]

for i in range(1, 11):
    poly = PolynomialFeatures(degree=i)
    x_train_poly = poly.fit_transform(x_train)
    x_test_poly = poly.fit_transform(x_test)

    clf = linear_model.LinearRegression()
    clf.fit(x_train_poly, y_train)
    print("The degree of the polynomial is {}".format(i))
    test_yhat = clf.predict(x_test_poly)
    train_yhat = clf.predict(x_train_poly)
    test_mse = mean_squared_error(y_test, test_yhat)
    train_mse = mean_squared_error(y_train, train_yhat)
    test_list.append(test_mse)
    train_list.append(train_mse)
    print("The test Accuracy is {}".format(test_mse))
    print("The training Accuracy is {}".format(train_mse))
    print('.....')


plt.figure(1)
plt.subplot(211)
plt.plot(train_list)
plt.ylabel('Training Error(MSE)')
plt.xlabel('Degree of the polynomial')
plt.grid(True)

plt.subplot(212)
plt.plot(test_list)
plt.ylabel('Testing Error(MSE)')
plt.xlabel('Degree of the polynomial')
plt.grid(True)
plt.show()
```

REFERENCES

[1] Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository http://archive.ics.uci.edu/ml. Irvine, CA: University of California, School of Information and Computer Science.

[2] P. Cortez and A. Morais. A Data Mining Approach to Predict Forest Fires using Meteorological Data. In J. Neves, M. F. Santos and J. Machado Eds., New Trends in Artificial Intelligence, Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence, December, Guimares, Portugal, pp. 512-523, 2007. APPIA, ISBN-13 978-989-95618-0-9.