GHARDA INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

Machine Learning Lab BE Computer (Semester-VII)

Experiment No.1: Linear Regression

Aim- To study, understand and implement a linear regression algorithm.

Theory

Linear Regression comes under the category of supervised machine learning algorithms. In supervised learning when given a data-set, we already know what the correct output should look like, we already have an idea of the relationship between the input and the output. Supervised learning broadly covers two types of problems:

1. Regression problems
2. Classification problems

In simple words, regression problems try to predict results within a continuous output i.e they try to map input variables to some continuous function. The output here is a continuous set. It also helps to remember that when the target variable we are trying to predict is continuous.

Simple Linear Regression: This is the simplest form of linear regression, and it involves only one independent variable and one dependent variable. The equation for simple linear regression is: $Y= \beta 0 + \beta 1.X$, where, Y is the dependent variable, X is the independent variable, $\beta 0$ is the intercept, $\beta 1$ is the slope

Scatter Plot: A scatter plot is a type of plot that displays values for two variables as points on a Cartesian plane. Each point represents a single observation with values for the variables plotted along the x-axis and y-axis. Scatter plots are useful for visually inspecting the relationship between two variables and identifying patterns or trends.

Regression Line: In statistics and machine learning, a regression line is a straight line that best fits the data points in a scatter plot. It represents the relationship between the independent variable (x-axis) and the dependent variable (y-axis) in a linear regression model. The regression line is typically expressed as: $y = mx + c$ where y) is the predicted value of the dependent variable, x is the independent variable, m is the slope of the line, which represents the rate of change of y with respect to x , c is the y-intercept, which is the value of y when x is 0.

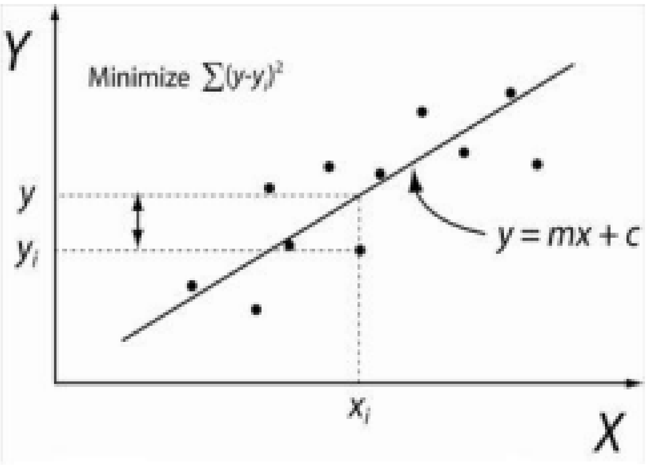


Fig.1:Linear Regression visual representation

Error in Prediction: In the context of regression analysis, error in prediction refers to the difference between the actual observed values of the dependent variable and the values predicted by the regression model. This difference is also known as the residual. The error in prediction indicates how well the regression model fits the data points on the scatter plot.

Best Fitting Line: The best fitting line (or best fit line) is the regression line that minimizes the sum of the squared differences between the observed values and the values predicted by the line. This method is known as the method of least squares. The best fitting line passes through the data points in such a way that the sum of the squared residuals (errors) is minimized.

In any supervised learning problem, our goal is simple:

> "Given a training set, we want to learn a function h: X →Y so that h(x) is a good prediction for the corresponding value of y"

Here h(x) is called the hypothesis function and is basically what we are trying to predict through our learning algorithm i.e. Linear Regression.

For the case of univariate simple linear regression our hypothesis function is:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

In the above equation, θ0 and θ1 are called the parameters of the hypothesis. It can be easily noticed that our equation for h(x) is actually just the mathematical equation for a line in a 2-dimensional plane and now we have seen that our hypothesis h(x) is in fact a line in the graphical sense as well hence the term "linear" regression.

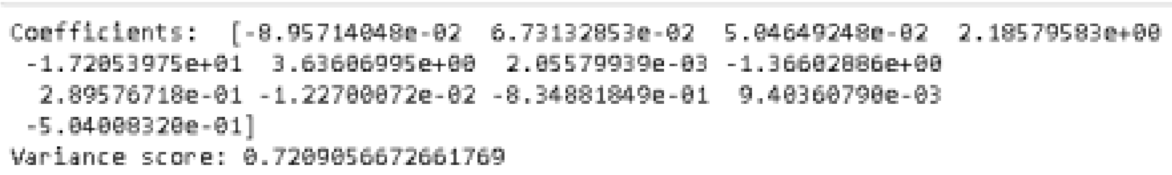
Cost Function: It's important to know how accurate our predictions are in order to know how well our model performs and if it needs further "training" or more "tuning" (which is basically adjustment of the parameters). This is where the cost function comes into the picture. The cost function is an expression through which we evaluate the quality of our current hypothesis and proceed to make changes accordingly. In simpler words, our cost function decides the cost we want our model to incur depending on how far off our predictions are from the true value. It is only intuitive to think that the "cost" should in fact be the difference between our prediction and the true value i.e. h(x)-y. The cost function for linear regression is:

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

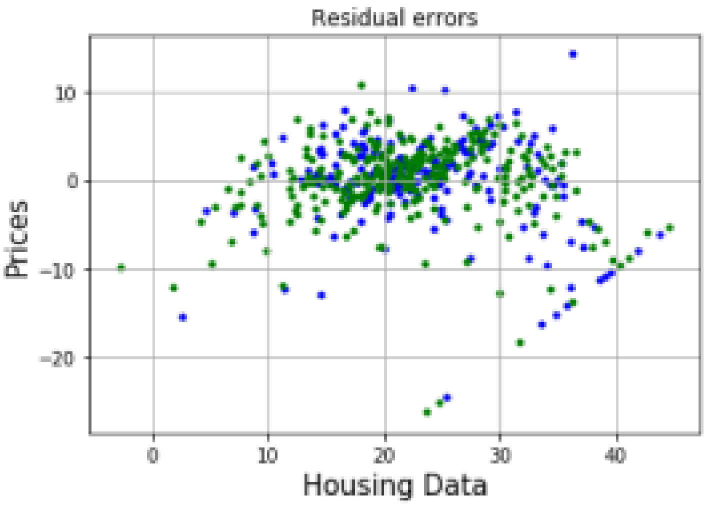
The main goal here is to minimize our cost function J(θ) so that we get h(x) as the function which passes through maximum points in the plot of X and Y or in other words we want to minimize the cost function so that the predictions of our model are as close as possible to the actual values.

Program Code and Output:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets, linear_model, metrics

from sklearn.datasets import fetch_california_housing

california_housing = fetch_california_housing()

X = california_housing.data # Features (X)
y = california_housing.target # Target (y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.4,random_state=1)

reg=linear_model.LinearRegression()

reg.fit(X_train,y_train)

print('Coefficients:',reg.coef_)

print('Variance score: {}'.format(reg.score(X_test,y_test)))

plt.scatter(reg.predict(X_test),reg.predict(X_test)-y_test,color="blue",s = 10,label = "Test data")

plt.scatter(reg.predict(X_train),reg.predict(X_train)-y_train,color="green",s = 10,label = "Train data")
#plt.hlines(y = 0,xmin = 0,xmax = 50,linewidth = 2)
#plt.legend(loc = "upper right")

plt.xlabel('x-axis', fontsize=20)
plt.ylabel('y-axis', fontsize=20)
plt.title("Residual errors")
plt.grid()
plt.show()
```



```
Coefficients:  [-8.95714048e-02  6.73132853e-02  5.04649248e-02  2.18579583e+00
 -1.72053975e+01  3.63606995e+00  2.05579939e-03 -1.36602886e+00
  2.89576718e-01 -1.22700072e-02 -8.34881849e-01  9.40360790e-03
 -5.04008320e-01]
Variance score: 0.7209056672661769
```



Conclusion:

The concept of linear regression is studied and implemented using least square method as well as python built-in functions with standard dataset.

References

1. http://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html

2. http://www.statisticssolutions.com/assumptions-of-linear-regression/ 3.
https://www.kaggle.com/datasets/camnugent/california-housing-prices