Machine Learning Using Python

LAB-2

Task : Understand Regression

Sol :

What Is Regression?

Regression searches for relationships among **variables**. For example, you can observe several employees of some company and try to understand how their salaries depend on their **features**, such as experience, education level, role, city of employment, and so on.

This is a regression problem where data related to each employee represents one **observation**. The presumption is that the experience, education, role, and city are the independent features, while the salary depends on them.

The dependent features are called the **dependent variables**, **outputs**, or **responses**. The independent features are called the **independent variables**, **inputs**, **regressors**, or **predictors**.

When Do You Need Regression?

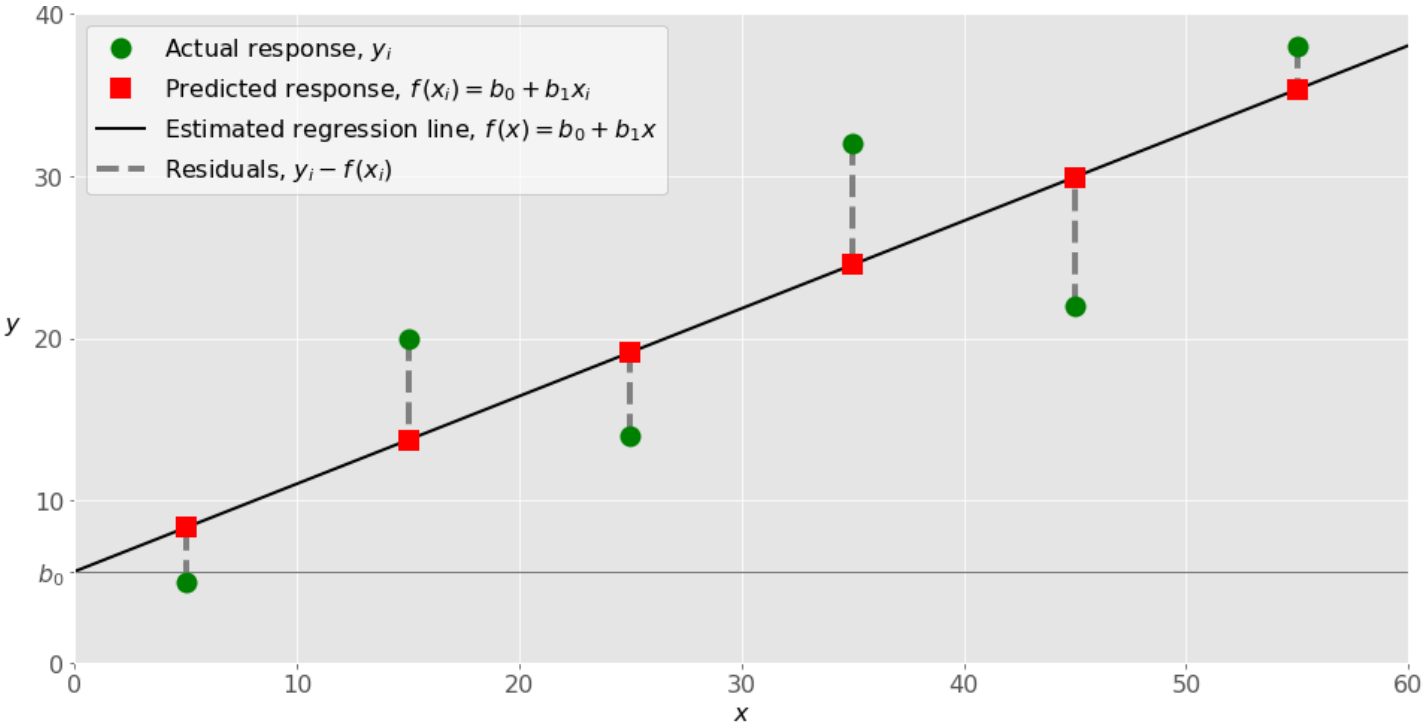
Regression is also useful when you want to **forecast** a response using a new set of predictors. For example, you could try to predict electricity consumption of a household for the next hour given the outdoor temperature, time of day, and number of residents in that household.

Linear Regression

Linear regression is probably one of the most important and widely used regression techniques. It’s among the simplest regression methods. One of its main advantages is the ease of interpreting results.

Simple Linear Regression

Simple or single-variate linear regression is the simplest case of linear regression, as it has a single independent variable, $\mathbf{x} = x$.



Example Code 1:

```
from sklearn import linear_model
features = [[2],[1],[5],[10]]
labels = [27, 11, 75, 155]
clf = linear_model.LinearRegression()
clf=clf.fit(features,labels)
#predicted = clf.predict([[5]])
predicted = clf.predict([[5], [5], [2]])
print(predicted)
```

Output : [75. 75. 27.]

Example Code 2 :

```
import matplotlib.pyplot as plt
from scipy import stats

x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

slope, intercept, r, p, std_err = stats.linregress(x, y) # use of Linerreg function

def myfunc(x):
  return slope * x + intercept

mymodel = list(map(myfunc, x))

plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()
```