## Finding the Regression Line – How Do We Optimize the Parameters of the Linear Model?

To find the regression line, we need to define the cost function, which is also called the loss function. Remember that the cost in machine learning is the error measure that the learning algorithm minimizes. We can also define the cost as the penalty when the model outputs an incorrect prediction.

In the case of the linear regression model, the cost function is defined as half the sum of the squared difference between the predicted value and the actual value. The linear regression cost function is called the ***squared error cost function*** and is written as

$$C(\theta)=\frac{1}{2}\Sigma\left(\hat{y}-y\right)^2$$

To put it more simply, the closer the approximate value of the target variable $\hat{y}$ is to the actual variable $y$, the lower our cost and the better our model.

Having defined the cost function, an optimization algorithm such as gradient descent is used to minimize the cost $C(\theta)$ by updating the weights of the linear regression model.

## How Do We Interpret the Linear Regression Model?

In machine learning, the focus of linear regression differs slightly from traditional statistics. In statistics, the goal of a regression model is to understand the relationships between the features and targets by interpreting p-values, whereas in machine learning, the goal of the linear regression model is to predict the targets given new samples.

Figure 19-4 shows a regression model with a line of best fit that optimizes the squared difference between the data features and the targets. This difference is also called the residuals (shown as the purple vertical lines in Figure 19-4). What we care about in a linear regression model is to minimize the error between the predicted labels and the actual labels in the dataset.

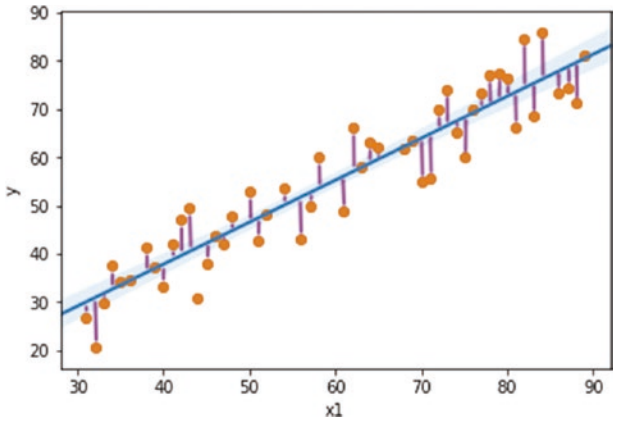

***Figure 19-4.*** *Linear regression model showing residuals*

If all the points in Figure 19-4 entirely fall on the predicted regression line, then the error will be 0. In interpreting the regression model, we want the error measure to be as low as possible.

However, our emphasis is to obtain a low error measure when we evaluate our model on the test dataset. Recall that the test of learning is when a model can generalize to examples that it was not exposed to during training.

## Linear Regression with Scikit-learn

In this example, we will implement a linear regression model with Scikit-learn. The model will predict house prices from the Boston house-prices dataset. The dataset contains 506 observations and 13 features.

We begin by importing the following packages:

```
sklearn.linear_model.LinearRegression: function that implements the
LinearRegression model.
sklearn.datasets: function to load sample datasets integrated with scikit-
learn for experimental and learning purposes.
sklearn.model_selection.train_test_split: function that partitions the
dataset into train and test splits.
sklearn.metrics.mean_squared_error: function to load the evaluation metric
for checking the performance of the model.
```