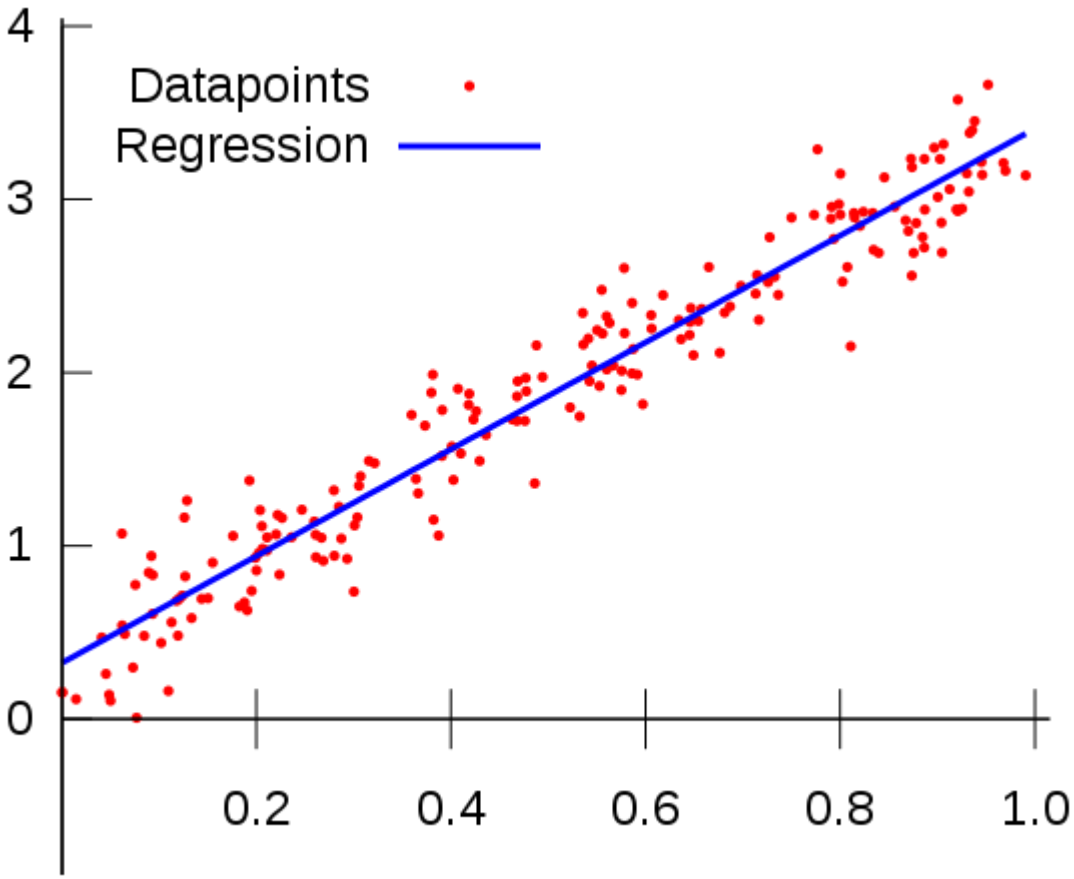
Regression

A regression problem is when we want to estimate the output of a continuous variable. In order to do so, we fit a function to the existing data and we use it for the generalization.



In order to do so, we start by defining the function with unknown parameters. For example, in the case above, we expect the mapping function to be linear (y = a*x + b).

Once defined, we minimize the cost function, which usually in this case is the sum of all the mean square errors between each training point, and the one given by the function.

Imports

In [1]:
```
from math import sqrt
import numpy as np
import matplotlib.pyplot as plt
from fit_plot import fit_plot
```

Regression toy problem

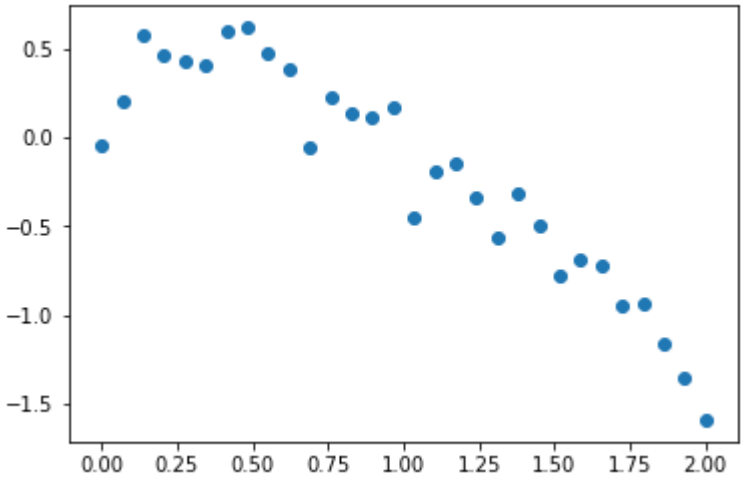
In [47]:
```
def true_fun(X):
    ''' Takes a 1D numpy array and returns a 1D numpy array,
    where f(x) = (1-x) * sqrt(x)'''
    return (1-X)*np.sqrt(X)
```

In [48]:
```
n_samples = 30
X = np.linspace(0,2,30)
Y = true_fun(X) + np.random.randn(n_samples)*0.15
```

Visualizing the data

In [49]:
```
plt.scatter(X, Y)
```

Out[49]:
```
<matplotlib.collections.PathCollection at 0x44ac1588>
```



We will use a function called 'fit_plot'.

This functin plots the true function, the sample, and the model found with the given degree of the polynomial.

In [53]:
```
fit_plot(X, Y, true_fun, [1,2,4])
```