

We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

[Continue](#)

Linear Regression using Gradient Descent



Adarsh Menon

[Follow](#)

Sep 16, 2018 · 5 min read

In this tutorial you can learn how the gradient descent algorithm works and implement it from scratch in python. First we look at what linear regression is, then we define the loss function. We learn how the gradient descent algorithm works and finally we will implement it on a given data set and make predictions.



We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

Continue

This is the equation for a line that you studied in high school. **m** is the slope of the line and **c** is the y intercept. Today we will use this equation to train our model with a given dataset and predict the value of **Y** for any given value of **X**. Our challenge today is to determine the value of **m** and **c**, such that the line corresponding to those values is the best fitting line or gives the minimum error.

Loss Function

The loss is the error in our predicted value of **m** and **c**. Our goal is to minimize this error to obtain the most accurate value of **m** and **c**.

We will use the Mean Squared Error function to calculate the loss. There are three steps in this function:

1. Find the difference between the actual y and predicted y value($y = mx + c$), for a given x.

2 We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on
3 September 1, 2020, and we encourage you to review them. By continuing to use our services,
you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

Continue

$$E = \frac{1}{n} \sum_{i=0}^n (y_i - \bar{y}_i)^2$$

Mean Squared Error Equation

Here y_i is the actual value and $\bar{y}_i$ is the predicted value. Lets substitute the value of $\bar{y}_i$:

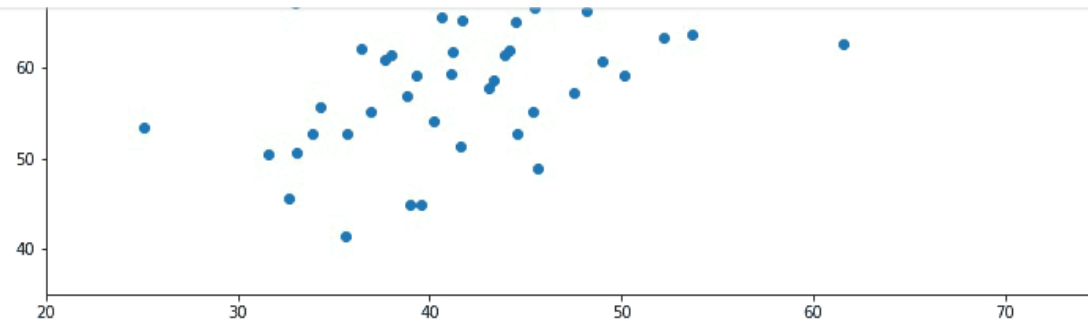
$$E = \frac{1}{n} \sum_{i=0}^n (y_i - (mx_i + c))^2$$

Substituting the value of $\bar{y}_i$

So we square the error and find the mean. hence the name Mean Squared Error. Now that we have defined the loss function, lets get into the interesting part — minimizing it and finding **m** and **c**.

We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

Continue



The values of m and c are updated at each iteration to get the optimal solution

Linear Regression using Gradient Descent in Python - Machine Learnin...



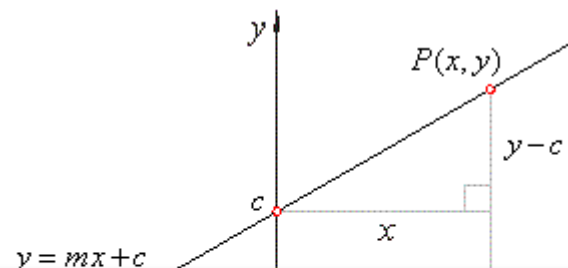
We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

Continue

Linear Regression

In statistics, linear regression is a linear approach to modelling the relationship between a dependent variable and one or more independent variables. Let X be the independent variable and Y be the dependent variable. We will define a linear relationship between these two variables as follows:

$$Y = mX + c$$



T
G
m

We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

Continue

Understanding Gradient Descent

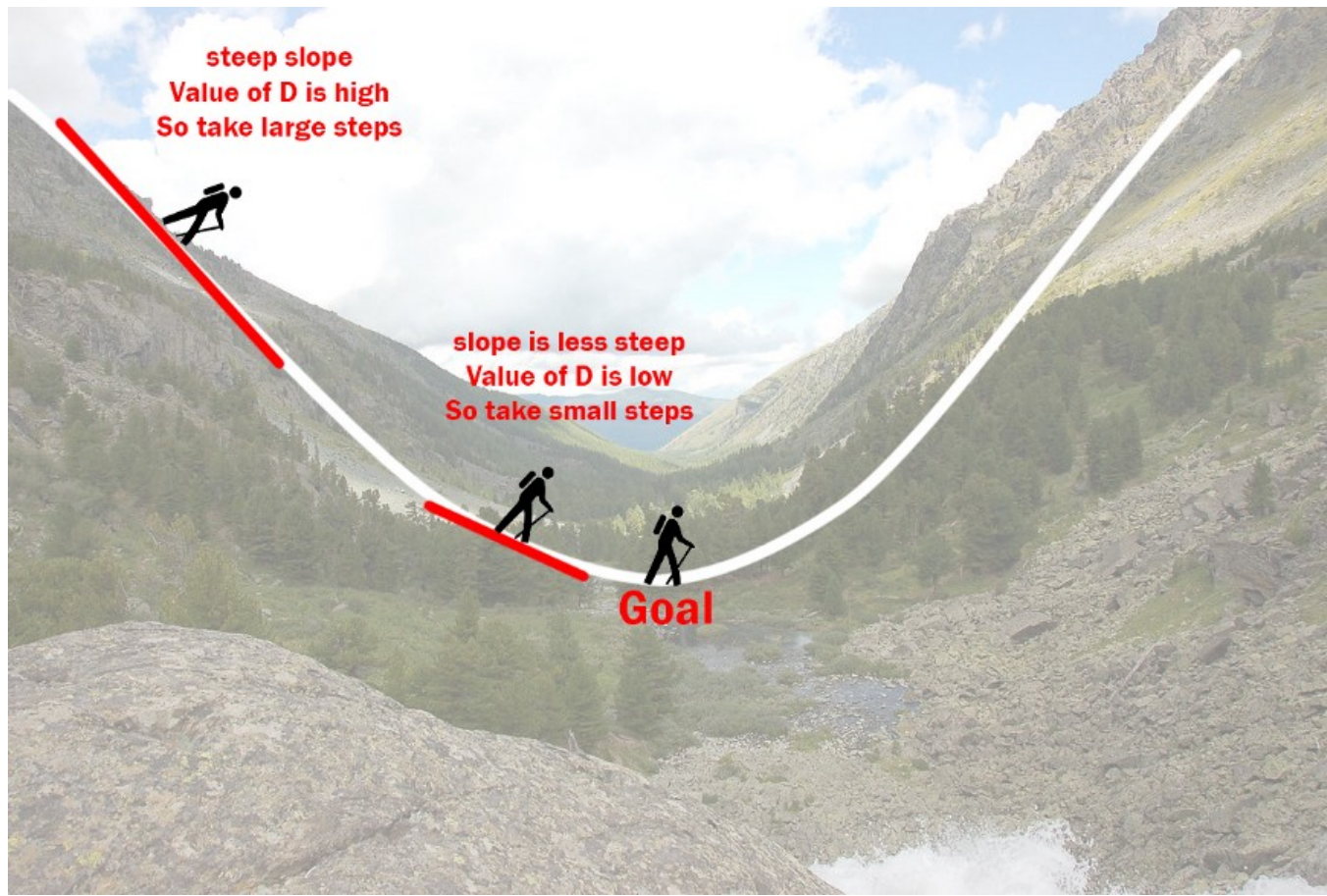


Illustration of how the gradient descent algorithm works

In
to
wl
de

We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

Continue

gets to the bottom of the valley which was his goal.

Let's try applying gradient descent to **m** and **c** and approach it step by step:

1. Initially let $m = 0$ and $c = 0$. Let L be our learning rate. This controls how much the value of **m** changes with each step. L could be a small value like 0.0001 for good accuracy.
2. Calculate the partial derivative of the loss function with respect to m , and plug in the current values of x , y , m and c in it to obtain the derivative value **D**.

$$D_m = \frac{1}{n} \sum_{i=0}^n 2(y_i - (mx_i + c))(-x_i)$$
$$D_m = \frac{-2}{n} \sum_{i=0}^n x_i(y_i - \bar{y}_i)$$

Derivative with respect to **m**

We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

Continue

$$D_c = \frac{-2}{n} \sum_{i=0}^n (y_i - \bar{y}_i)$$

Derivative with respect to **c**

3. Now we update the current value of **m** and **c** using the following equation:

$$m = m - L \times D_m$$

$$c = c - L \times D_c$$

4. We repeat this process until our loss function is a very small value or ideally 0 (which means 0 error or 100% accuracy). The value of **m** and **c** that we are left with now will be the optimum values.

No We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

Continue

of the steps he will take. When the slope is more steep (D is more) he takes longer steps and when it is less steep (D is less), he takes smaller steps. Finally he arrives at the bottom of the valley which corresponds to our loss $= 0$.

Now with the optimum value of m and c our model is ready to make predictions !

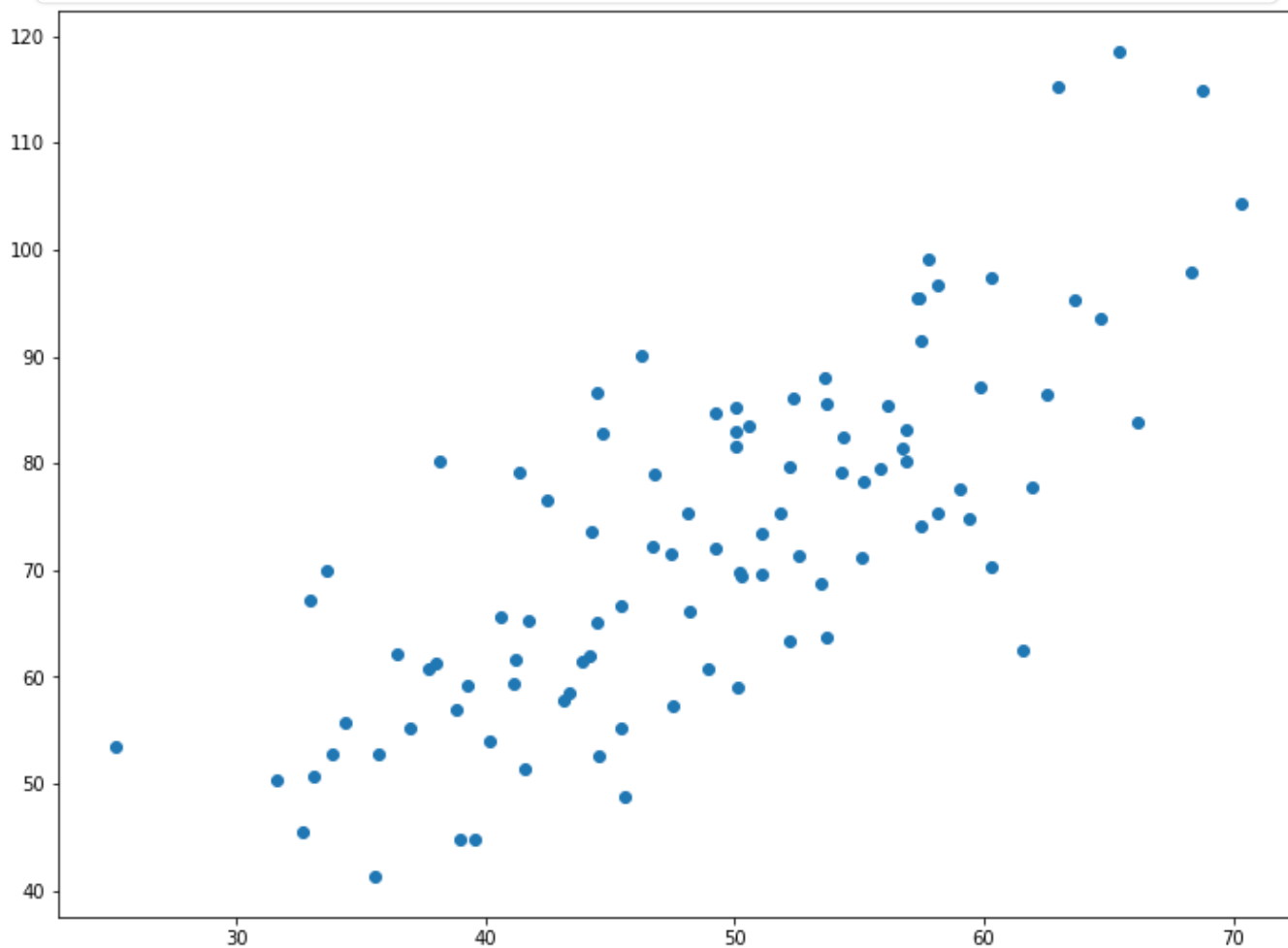
Implementing the Model

Now let's convert everything above into code and see our model in action !

```
1 # Making the imports
2 import numpy as np
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 plt.rcParams['figure.figsize'] = (12.0, 9.0)
6
7 # Preprocessing Input data
8 data = pd.read_csv('data.csv')
9 X = data.iloc[:, 0]
10 Y = data.iloc[:, 1]
11 plt.scatter(X, Y)
```

1 We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on
September 1, 2020, and we encourage you to review them. By continuing to use our services,
you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

Continue



```
1 # Building the model
2 m = 0
3 c = 0
```

We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

[Continue](#)

```
10 # Performing Gradient Descent
11 for i in range(epochs):
12     Y_pred = m*X + c # The current predicted value of Y
13     D_m = (-2/n) * sum(X * (Y - Y_pred)) # Derivative wrt m
14     D_c = (-2/n) * sum(Y - Y_pred) # Derivative wrt c
15     m = m - L * D_m # Update m
16     c = c - L * D_c # Update c
17
18 print (m, c)
```

linear_regression_gd2.py hosted with ❤ by GitHub

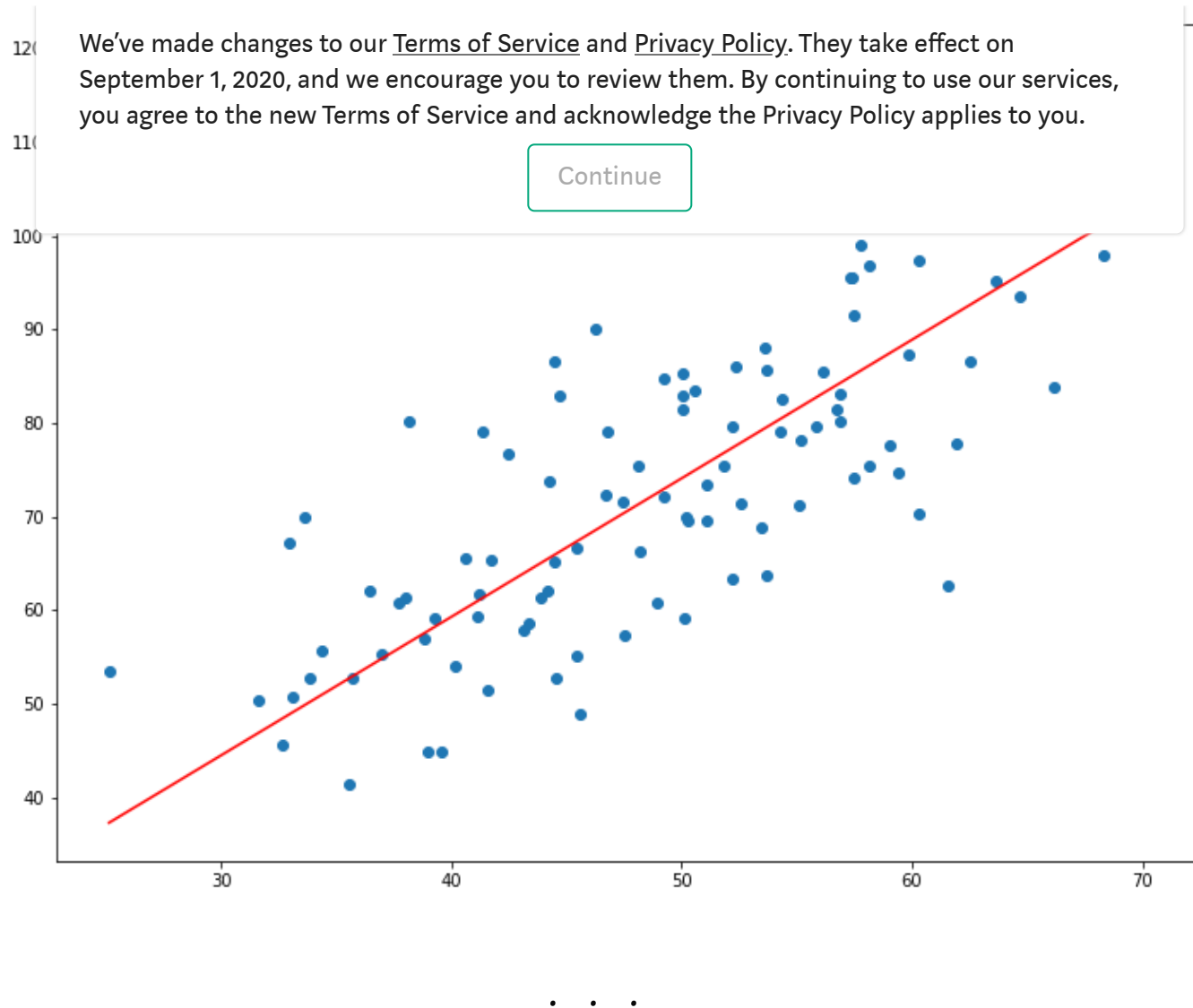
[view raw](#)

1.4796491688889395 0.10148121494753726

```
1 # Making predictions
2 Y_pred = m*X + c
3
4 plt.scatter(X, Y)
5 plt.plot([min(X), max(X)], [min(Y_pred), max(Y_pred)], color='red') # regression line
6 plt.show()
```

linear_regression_gd3.py hosted with ❤ by GitHub

[view raw](#)



Gradient descent is one of the simplest and widely used algorithms in machine learning, mainly because it can be applied to any function to optimize it. Learning it lays the foundation to mastering machine learning.

Fi
fr
SC
%

We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

Continue

Got questions ? Need help ? Contact me!

Email: adarsh1021@gmail.com

LinkedIn: <https://www.linkedin.com/in/adarsh-menon-739573146/>

Twitter: https://twitter.com/adarsh_menon_

Instagram: https://www.instagram.com/adarsh_menon_/

References:

Gradient Descent For Machine Learning

Optimization is a big part of machine learning. Almost every machine learning algorithm has an optimization algorithm...

machinelearningmastery.com

We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

[Continue](#)

Sign up for The Daily Pick

By Towards Data Science

Hands-on real-world examples, research, tutorials, and cutting-edge techniques delivered Monday to Thursday. Make learning your daily ritual. [Take a look](#)

Your email

[Get this newsletter](#)

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

[Machine Learning](#)[Gradient Descent](#)[Linear Regression](#)[Beginner](#)[Tutorial](#)

Discover Medium

Welcome to a place where words matter.
On Medium, smart voices and original ideas

Make Medium yours

Follow all the topics you care about, and
we'll deliver the best stories for you to your
homepage and inbox. Explore

Become a member

Get unlimited access to the best stories on
Medium — and support writers while you're
at it. Just \$5/month. Upgrade

take center stage -
Watch

We've made changes to our [Terms of Service](#) and [Privacy Policy](#). They take effect on September 1, 2020, and we encourage you to review them. By continuing to use our services, you agree to the new Terms of Service and acknowledge the Privacy Policy applies to you.

Continue

About

Help

Legal