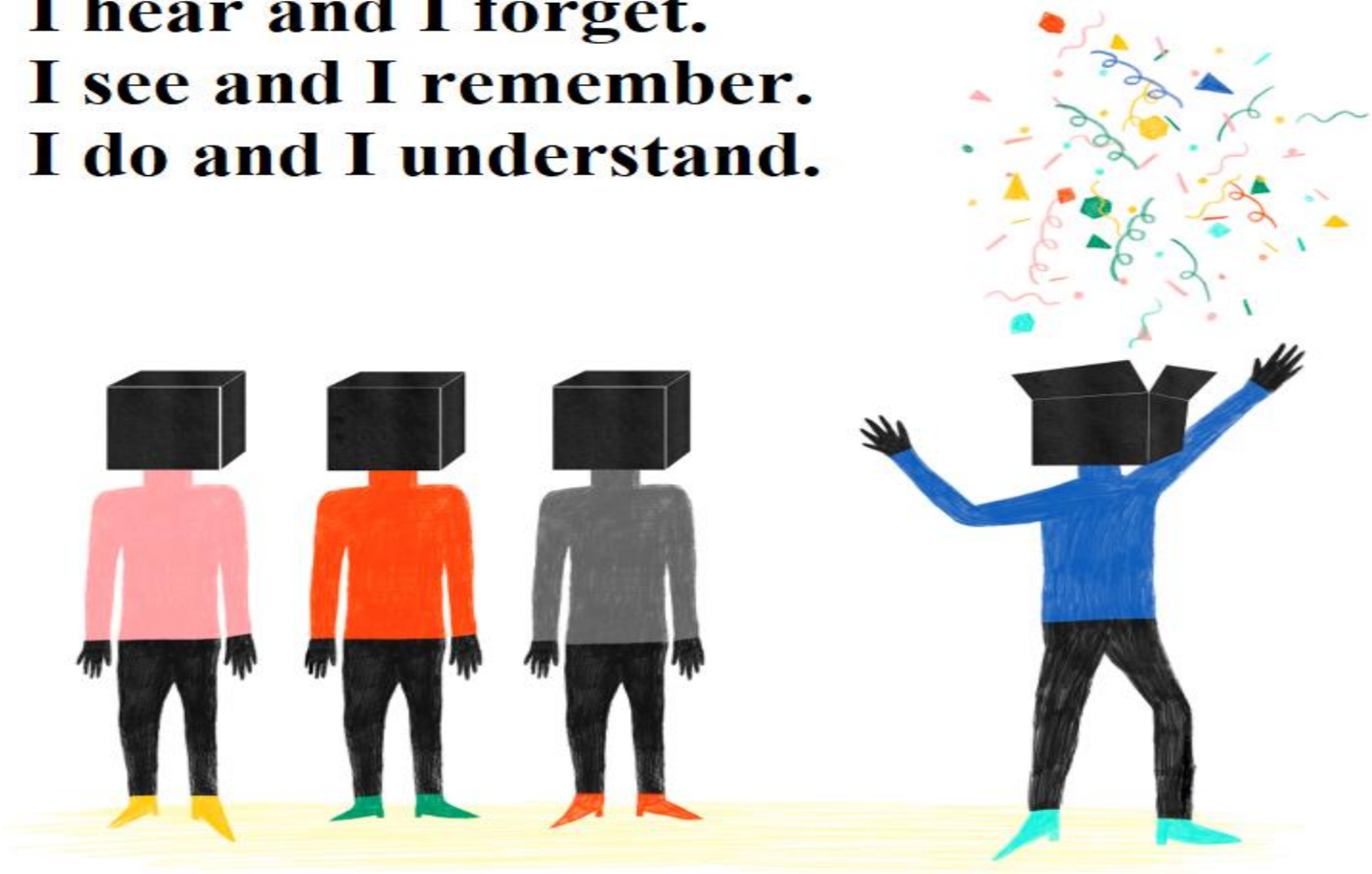
I hear and I forget.
I see and I remember.
I do and I understand.

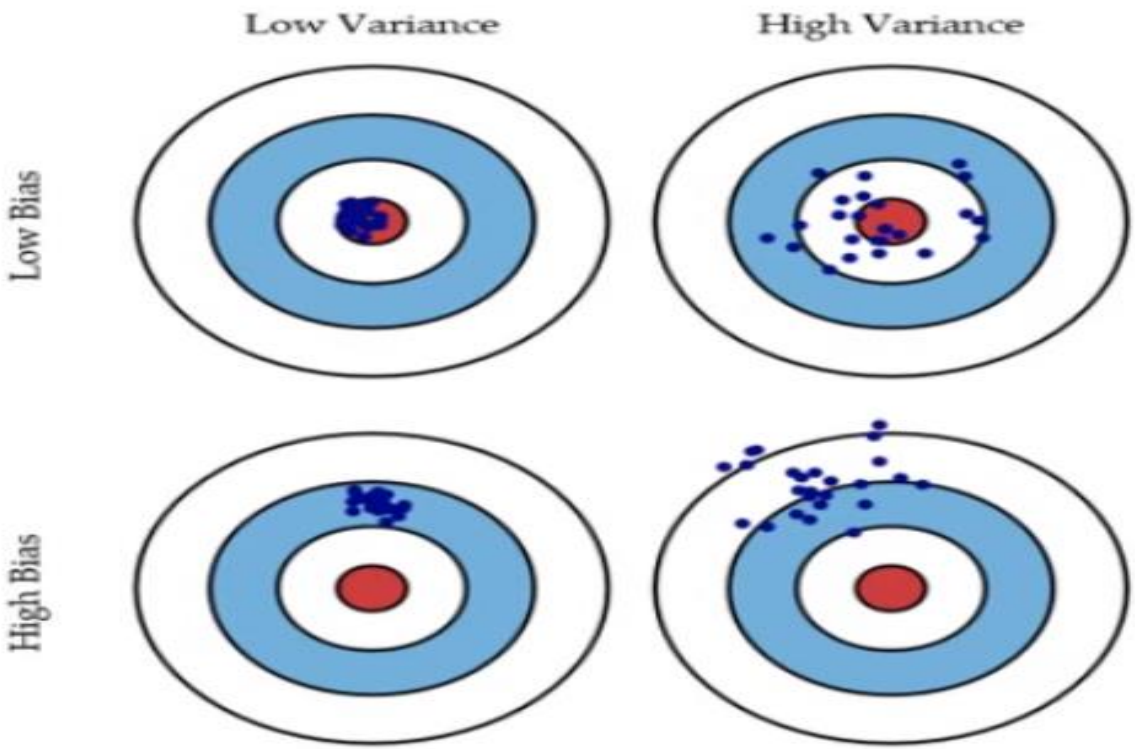


Chandan Verma

Corporate Trainer(Machine Learning,AI,Cloud Computing,IOT)

www.facebook.com/verma.chandan.070

Bias and Variance in regression models



Bias and variance is balanced to have a perfect model

Bias-Variance

High variance problem occurs when our model is able to predict well in our training dataset but fails on the test set. In such a case, we say **our model is not able to generalize well** and has **overfitted** to the training data

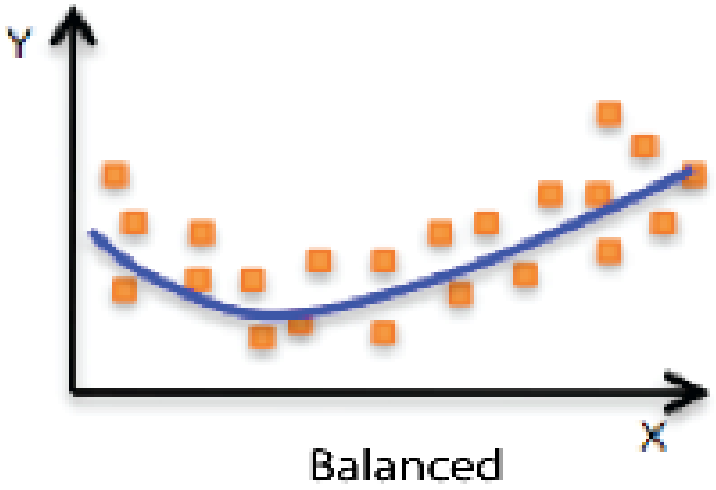
if we choose a simple model to predict a complex pattern, our model will have predictions far from actual values, it's known as **underfitting** or **high bias problem**.

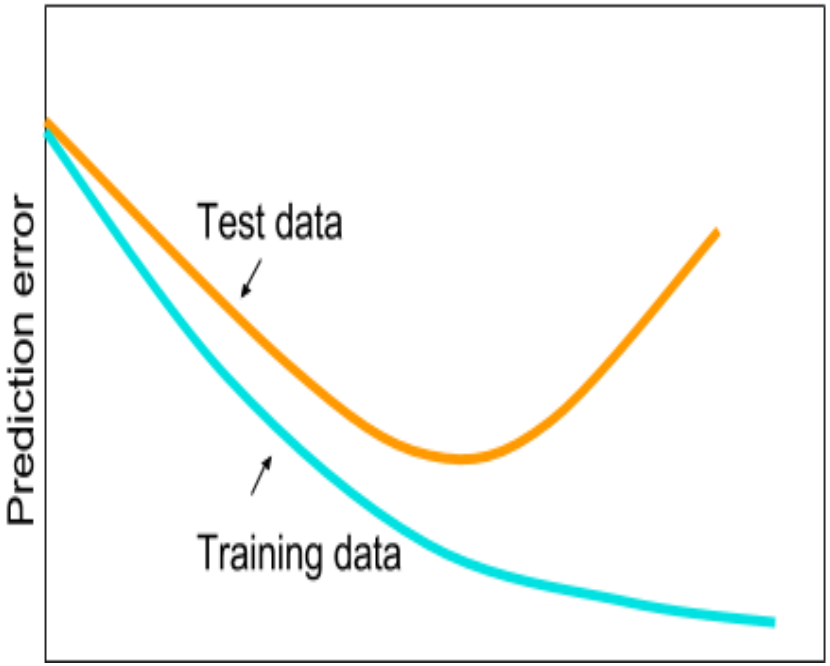
So, if we choose a more complicated algorithm, we run a risk of high variance problem while if we use a simple one, we will face high bias problem.

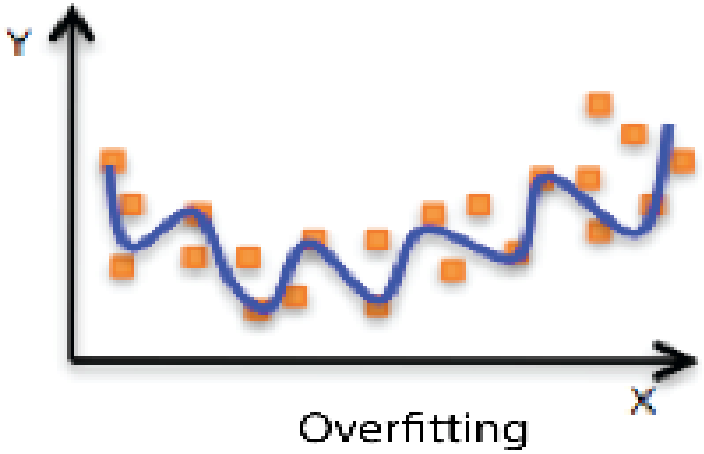
Over-fitting

overfitting happens when model learns signal as well as noise in the training data and wouldn't perform well on new data on which model wasn't trained on.

model that fits our training data well but fails to estimate the real relationship among variables beyond the training set. Therefore our model performs poorly on the test data





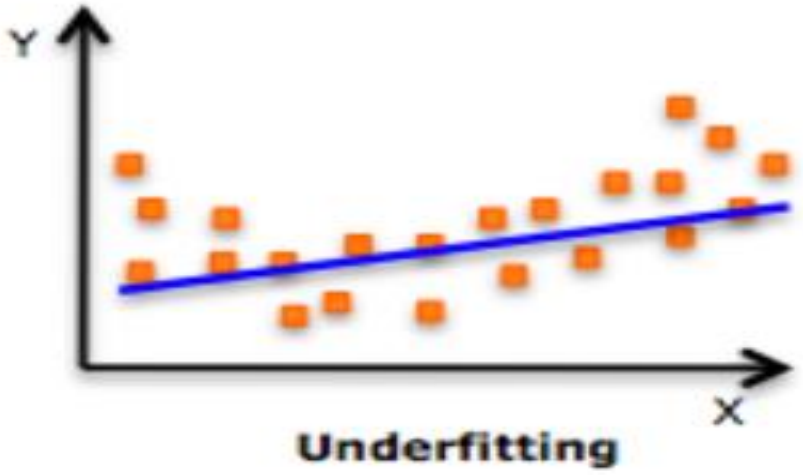


Model has high variance and low bias

Underfitting

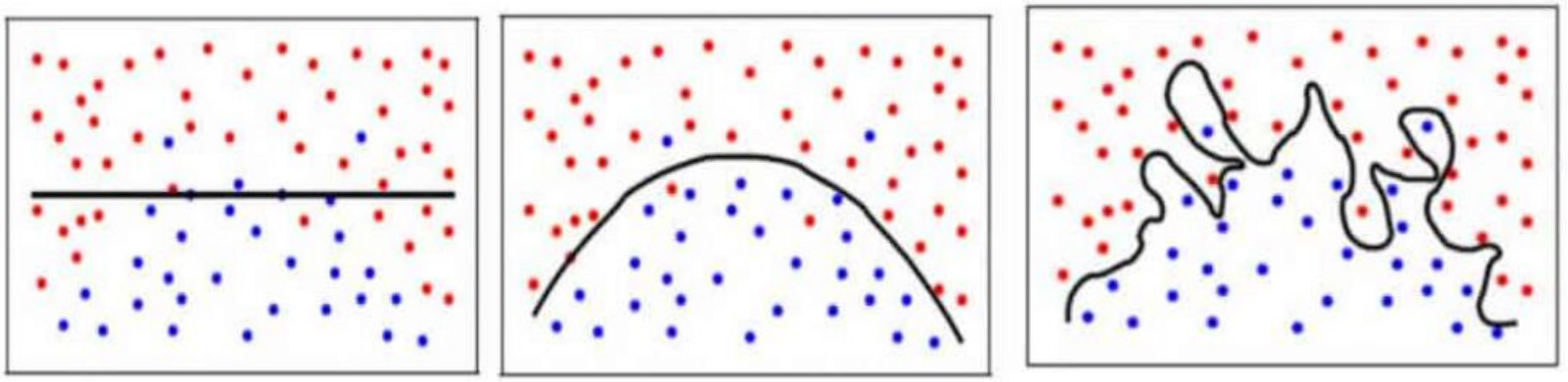
it occurs when our model neither fits the training data nor generalizes on the new data

if we choose a simple model to predict a complex pattern, our model will have predictions far from actual values, it's known as **underfitting** or **high bias problem**.



Bias-variance trade-off

In supervised machine learning, there is always a trade-off between approximation and generalization, known as **bias-variance trade-off**.

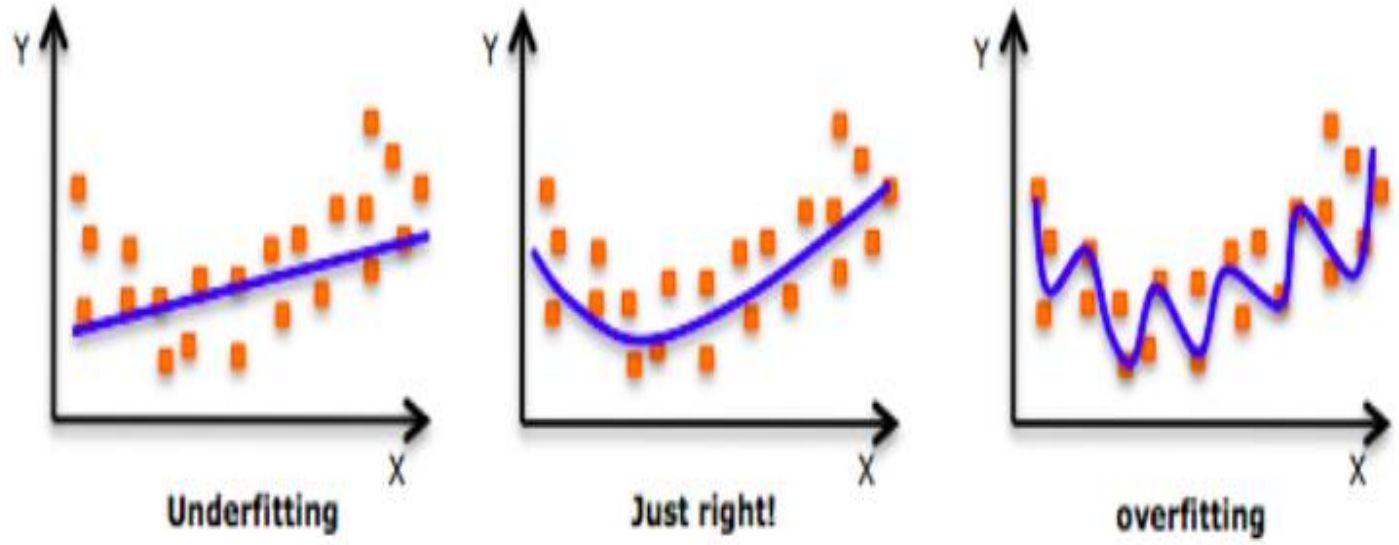


Avoid Overfitting - Underfitting

To overcome underfitting or high bias, we can basically add new parameters to our model so that the model complexity increases, and thus reducing high bias.

Now, there are few ways you can avoid overfitting your model on training data like cross-validation sampling, reducing number of features, pruning, regularization etc

Regularization

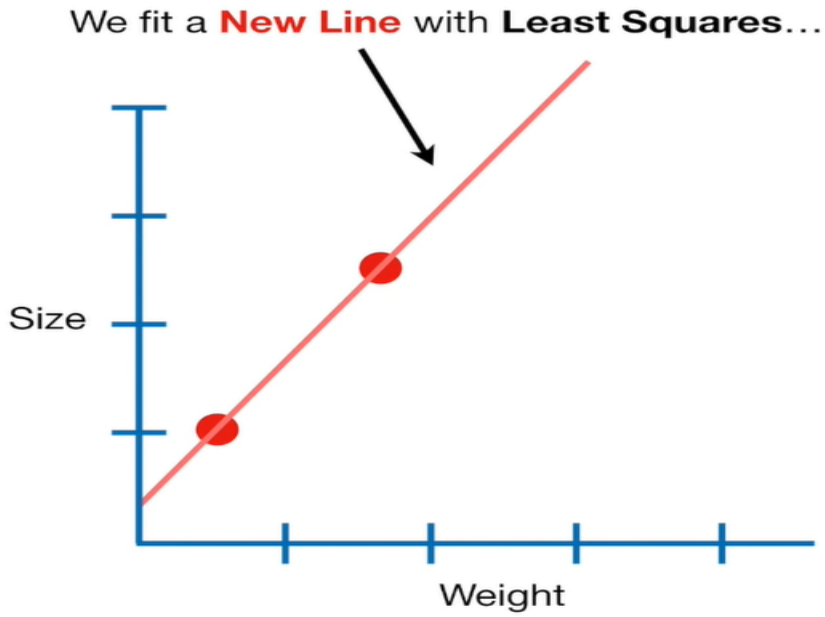


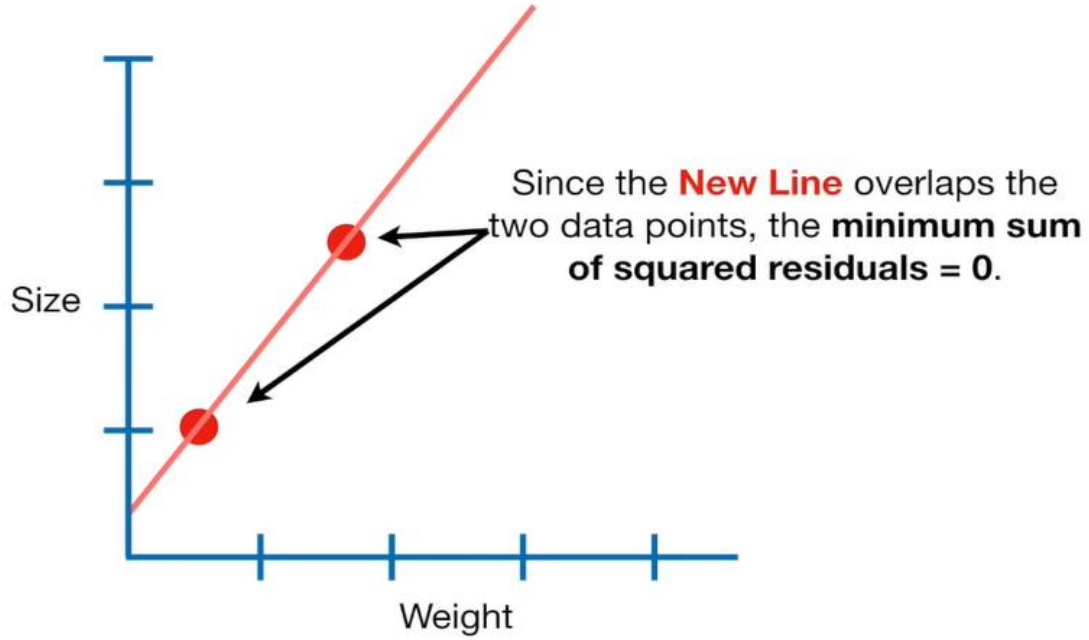
One of the most common problem data science professionals face is to avoid overfitting. Avoiding overfitting can single-handedly improve our model's performance.

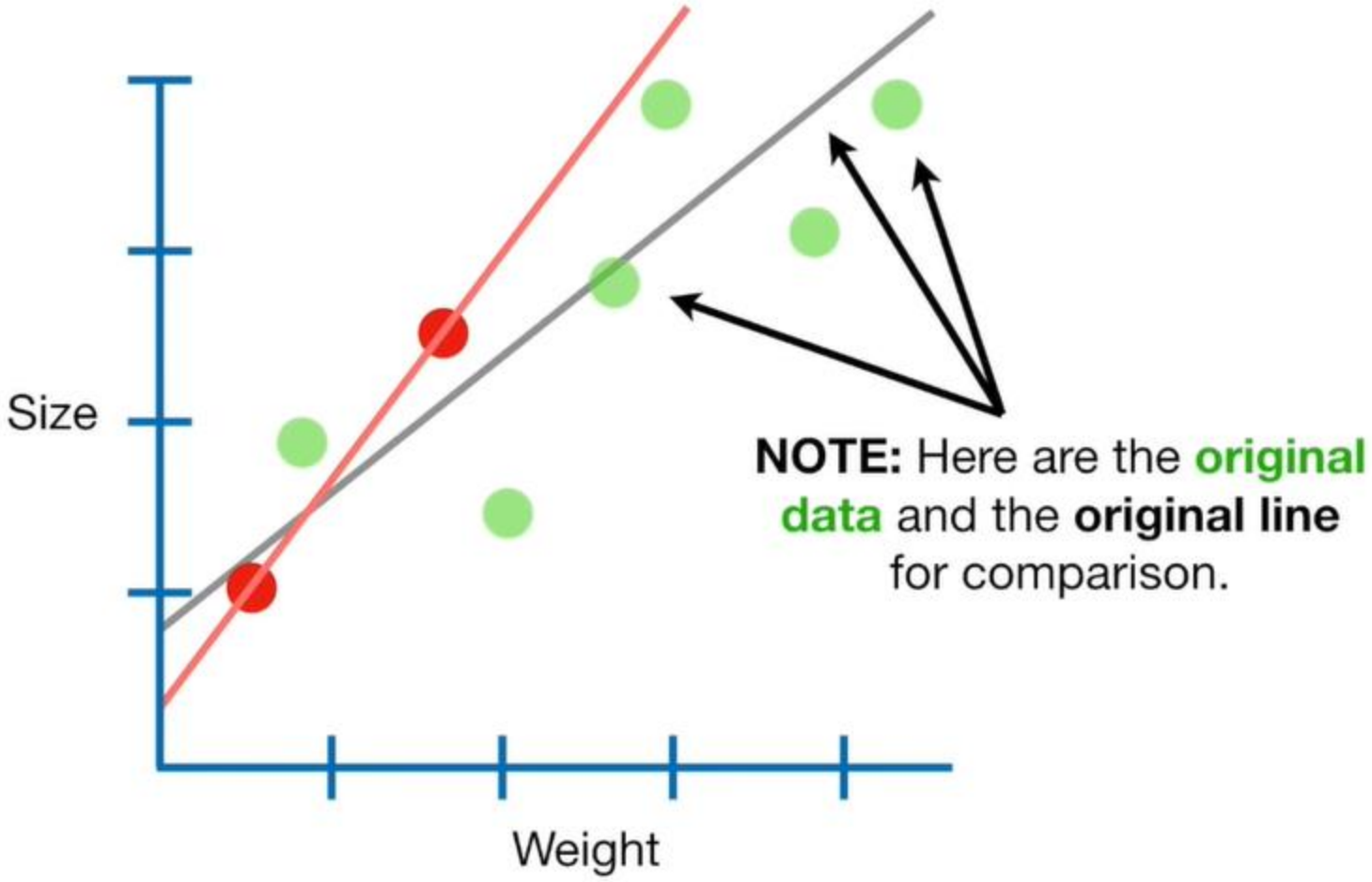
sum of squared errors (SSE)

The objective of ordinary least squares regression is to find the plane that minimizes the sum of squared errors (SSE) between the observed and predicted response.

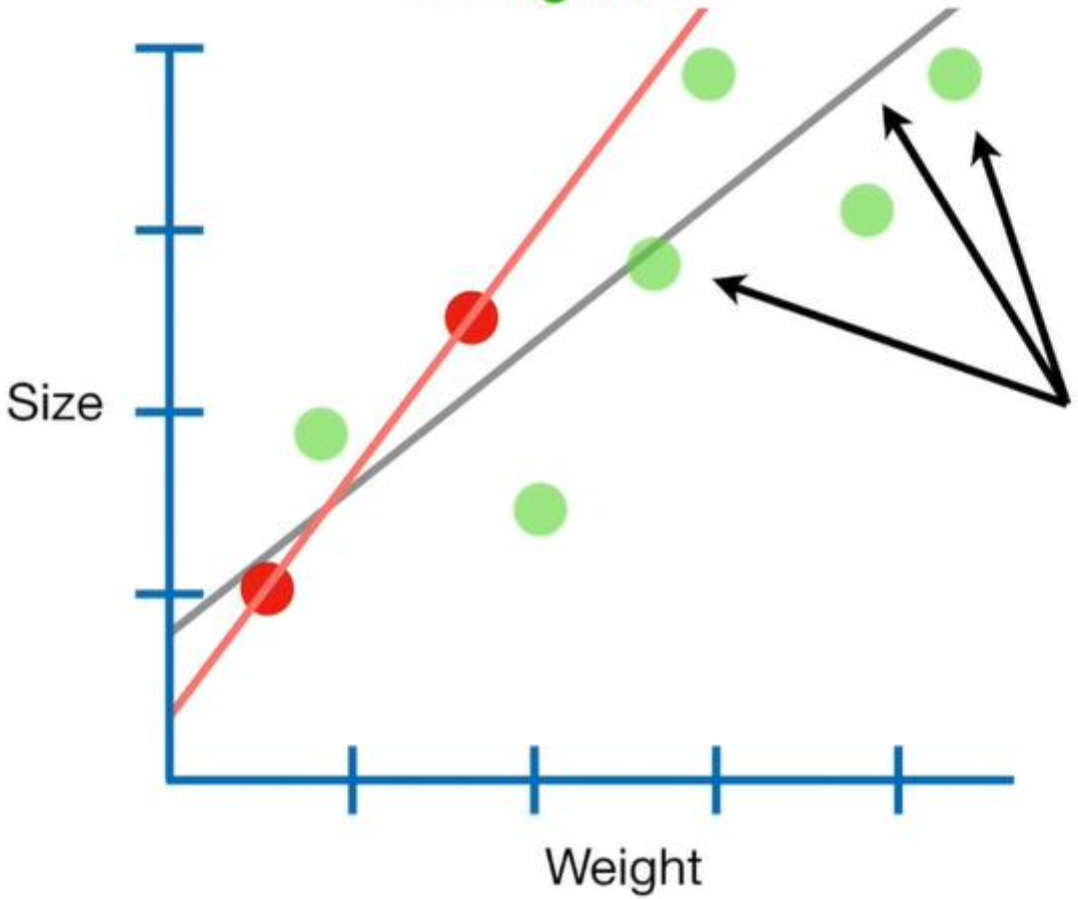
$$\text{minimize}\left\{ SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \right\}$$


We fit a New Line with Least Squares...
Size
Weight

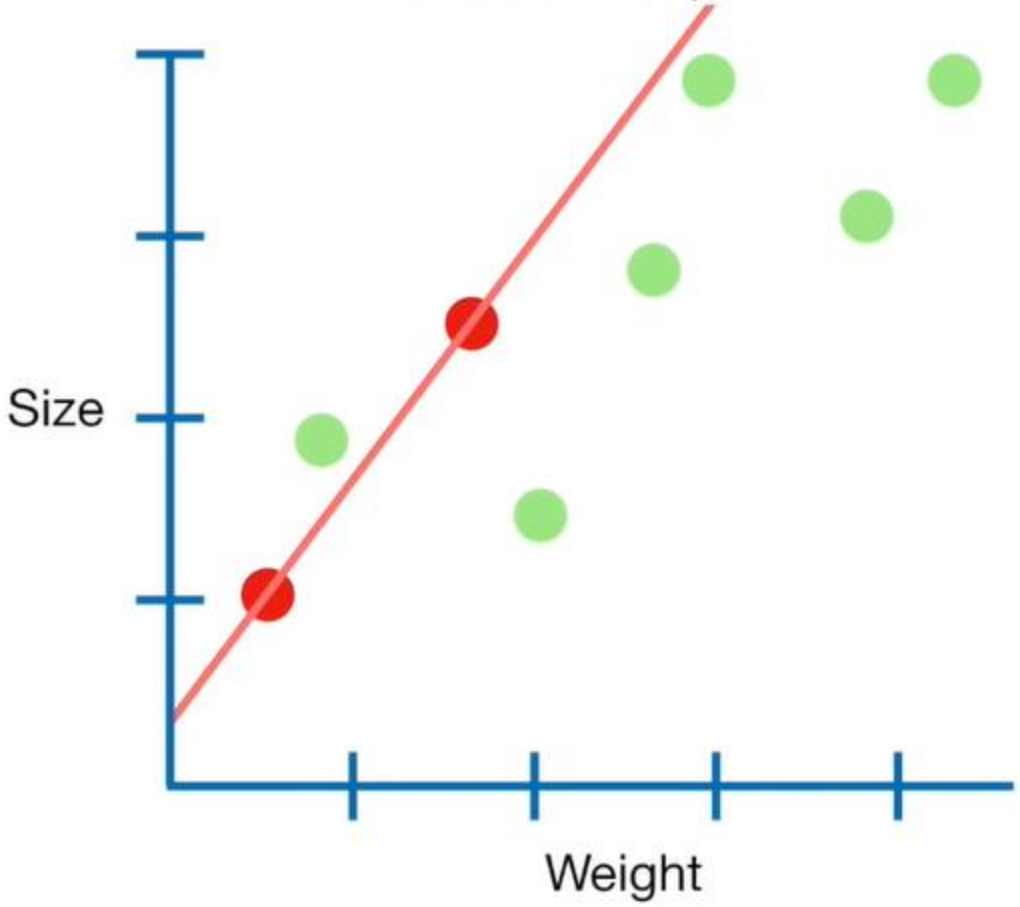

Since the New Line overlaps the two data points, the minimum sum of squared residuals = 0.
Size
Weight


Size
Weight
NOTE: Here are the original data and the original line for comparison.

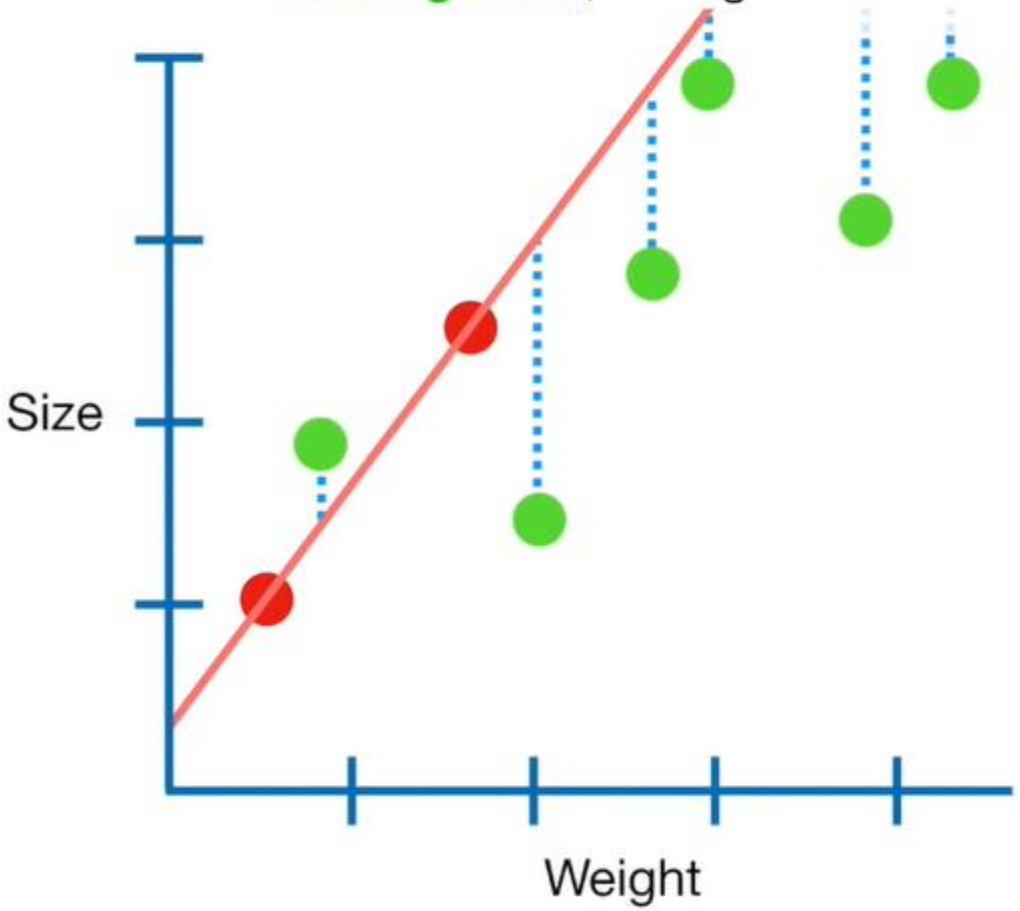
Let's call the **Two Red Dots** the **Training Data**, and the remaining **Green Dots** the **Testing Data**.



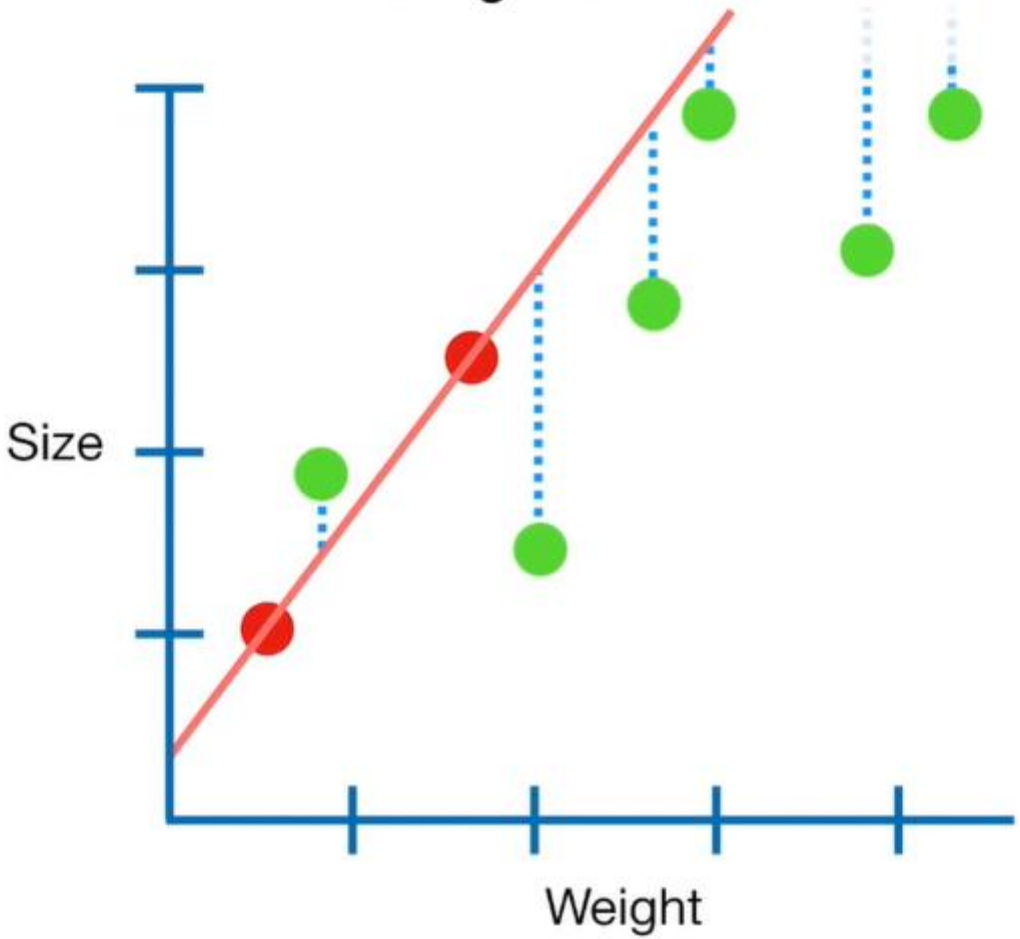
The sum of the squared residuals for just the **Two Red Points**, the **Training Data**, is small (in this case it is 0)…



…but the sum of the squared residuals for the **Green Points**, the **Testing Data**, is large…



…and that means that the **New Line** has **High Variance**.



Gradient Descent

$$Loss\ Function\ (LF) = \frac{1}{N}\sum(y - (mx + c))^2$$

$$\frac{\partial}{\partial m}LF = \frac{2}{N}\sum(y - (mx + c)) \times x$$

$$\frac{\partial}{\partial c}LF = \frac{2}{N}\sum(y - (mx + c))$$

key assumptions of OLS regression

1. Linear relationship
2. Multivariate normality
3. No autocorrelation
4. Homoscedastic (constant variance in residuals)
5. There are more observations (n) than features (p)
6. No or little multicollinearity

Features Increases

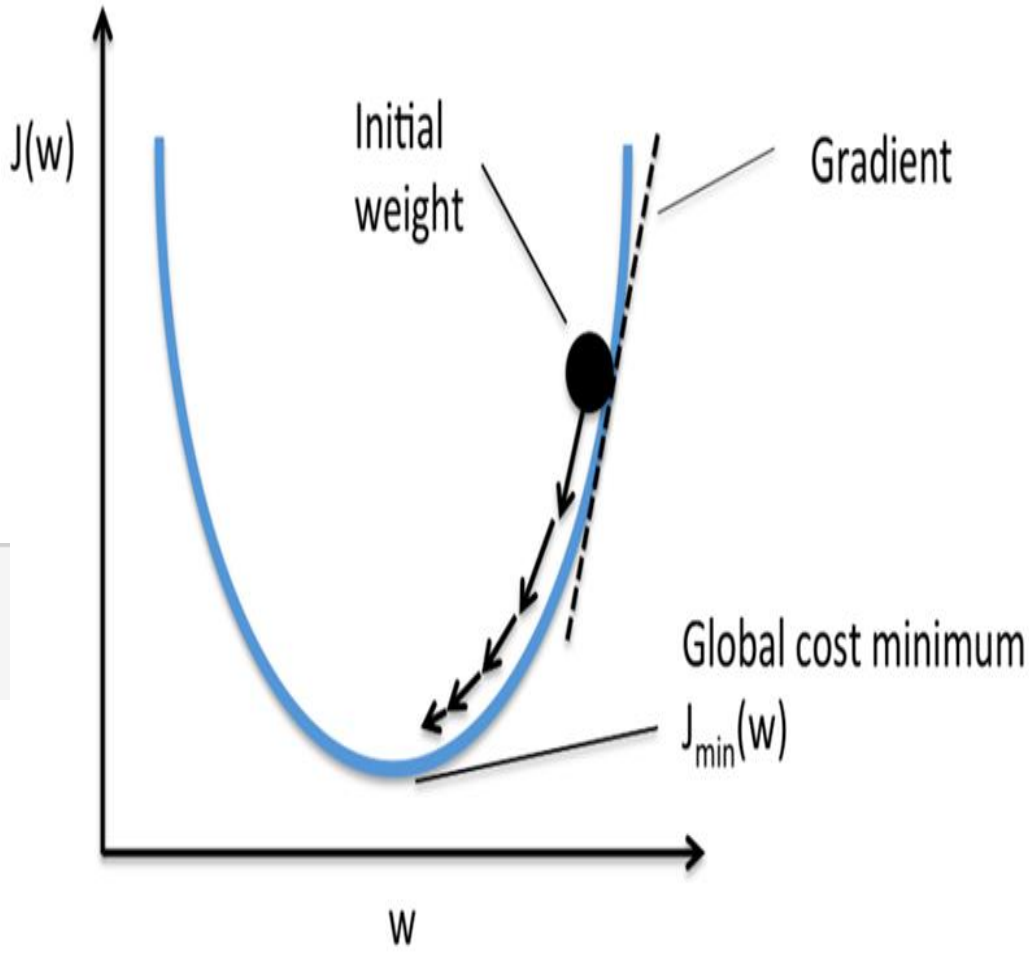
- Multicollinearity(high variance)
- Insufficient Solution
- Interpretability

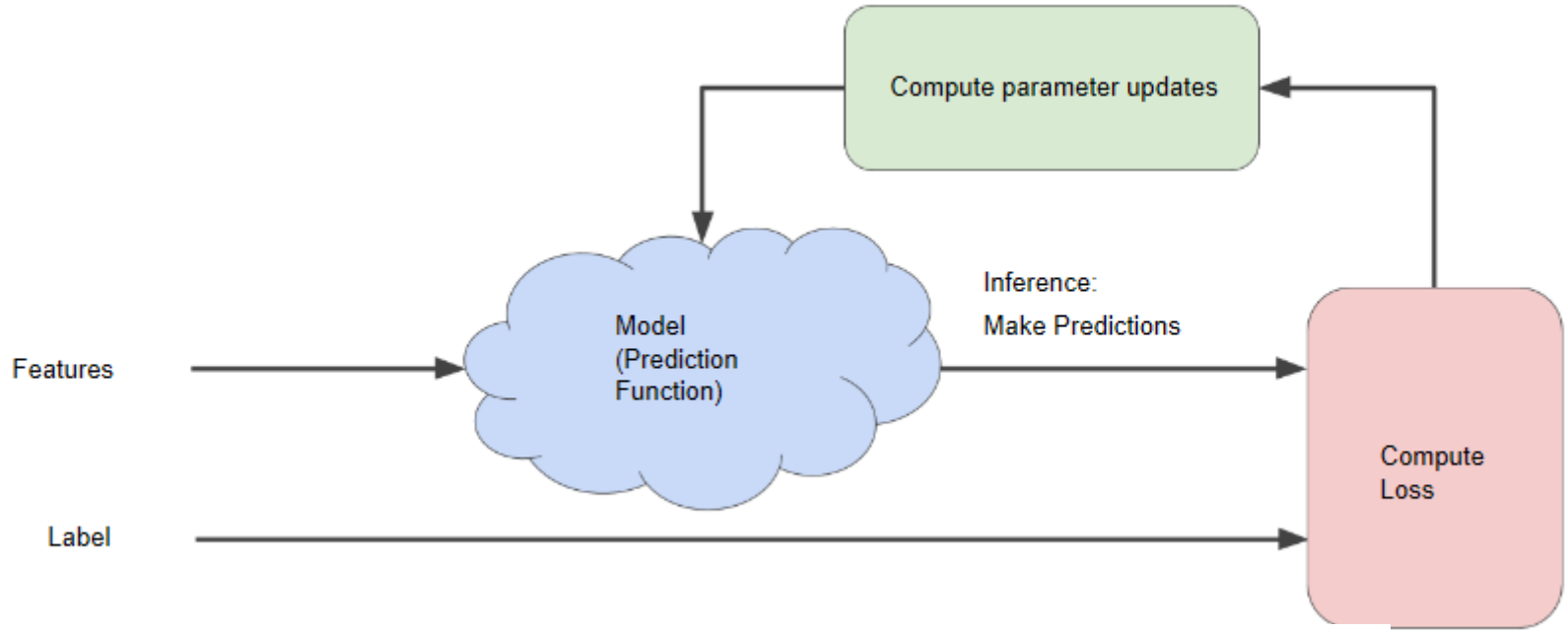
Iterative Calculation

Repeat until converge

$$m_{n+1} = m_n - \alpha \frac{\partial}{\partial m_n} LF(m_n)$$

$$c_{n+1} = c_n - \alpha \frac{\partial}{\partial c_n} LF(c_n)$$



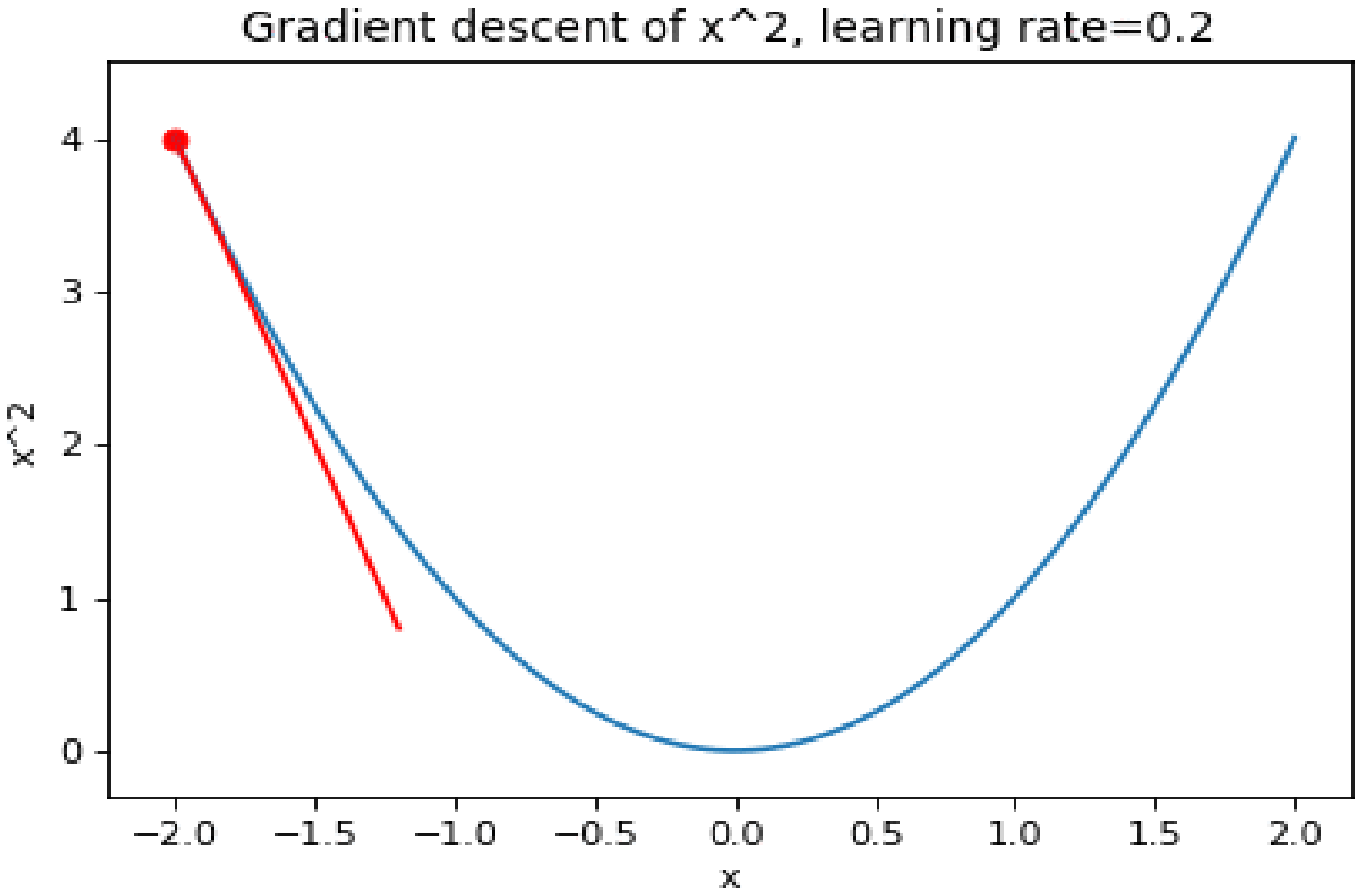


repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$$

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

}


Gradient descent of x^2, learning rate=0.2
x^2
x
4
3
2
1
0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

Regularization

Reducing overfitting in polynomial models is through the use of regularization

Regularization methods provide a means to control our regression coefficients, which can reduce the variance and decrease our of sample error.

commonly referred to as *penalized* models or *shrinkage* methods

Regularization

The objective function of regularized regression methods is very similar to OLS regression; however, we add a penalty parameter (P)

$$\text{minimize}\{SSE + P\}$$

Regularization basically adds the penalty as model complexity increases. Regularization parameter (lambda) penalizes all the parameters except intercept so that model generalizes the data and won't overfit.

The regularization techniques are as follows

1. Penalize the magnitude of coefficients of features
2. Minimize the error between the actual and predicted observations

Regularized regression puts constraints on the magnitude of the coefficients and will progressively shrink them towards zero.

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

$$\min_\theta J(\theta)$$

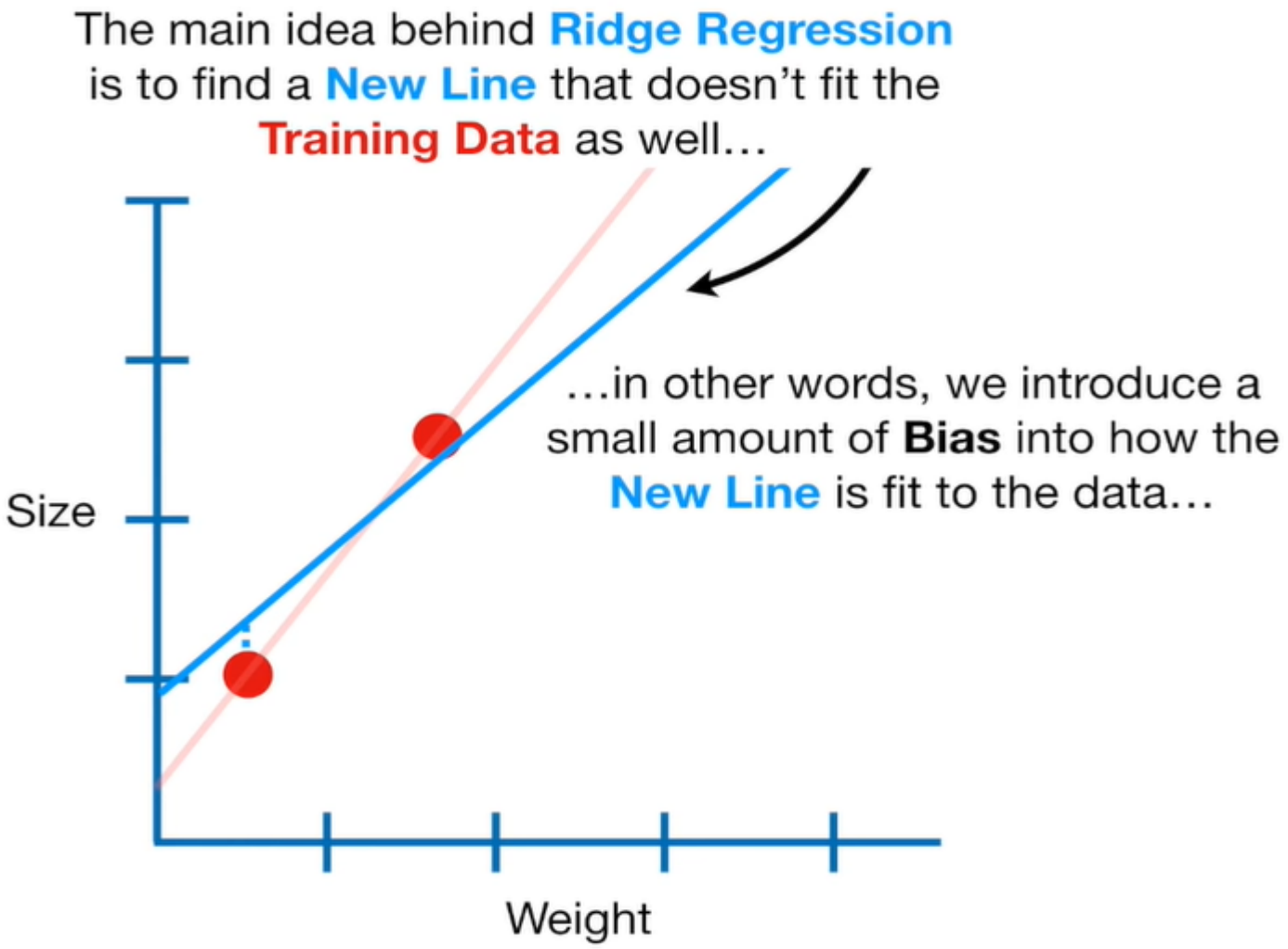
L2 Regularization: Ridge Regression

In ridge regression, the cost function is altered by adding a penalty equivalent to square of the magnitude of the coefficient

The Ridge regression is a technique which is specialized to analyze multiple regression data which is multicollinearity in nature.

The concept multicollinearity occurs when there are high co-relations between more than two predicted variables.

L2 Regularization: Ridge Regression



Correcting overfitting with regularization

Regularization adds a penalty on the different parameters of the model to reduce the freedom of the model. Hence, the model will be less likely to fit the noise of the training data and will improve the generalization abilities of the model.

- The L1 regularization (also called Lasso)
- The L2 regularization (also called Ridge)
- The L1/L2 regularization (also called Elastic net)

L2 Regularization (Ridge penalisation)

The L2 regularization adds a penalty equal to the sum of the squared value of the coefficients.

The L2 regularization will force the parameters to be relatively small, the bigger the penalization, the smaller (and the more robust) the coefficients are

As ridge regression shrinks the coefficients towards zero, it introduces some bias. But it can reduce the variance to a great extent which will result in a better mean-squared error. The amount of shrinkage is controlled by λ which multiplies the ridge penalty. As large λ means more shrinkage, we can get different coefficient estimates for the different values of λ.

L2 Regularization: Ridge Regression

The minimization objective =
LS Obj + λ (sum of the square of coefficients)

Here λ is the turning factor that controls the strength of the penalty term.
If $\lambda = 0$, the objective becomes similar to simple linear regression. So we get the same coefficients as simple linear regression.
If $\lambda = \infty$, the coefficients will be zero because of infinite weightage on the square of coefficients as anything less than zero makes the objective infinite.
If $0 < \lambda < \infty$, the magnitude of λ decides the weightage given to the different parts of the objective.

L1 Regularization (Lasso penalisation)

The L1 regularization adds a penalty equal to the sum of the absolute value of the coefficients.

Least Absolute Shrinkage and Selection Operator adds absolute value of Magnitude of Coefficient as penalty term to the loss function

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Cost function