Lecture 6 – Training, Test & Validation

ME494 – Data Science and Machine Learning for Mech Engg

Instructor – Subramanian Sankaranarayanan

Teaching Assistants

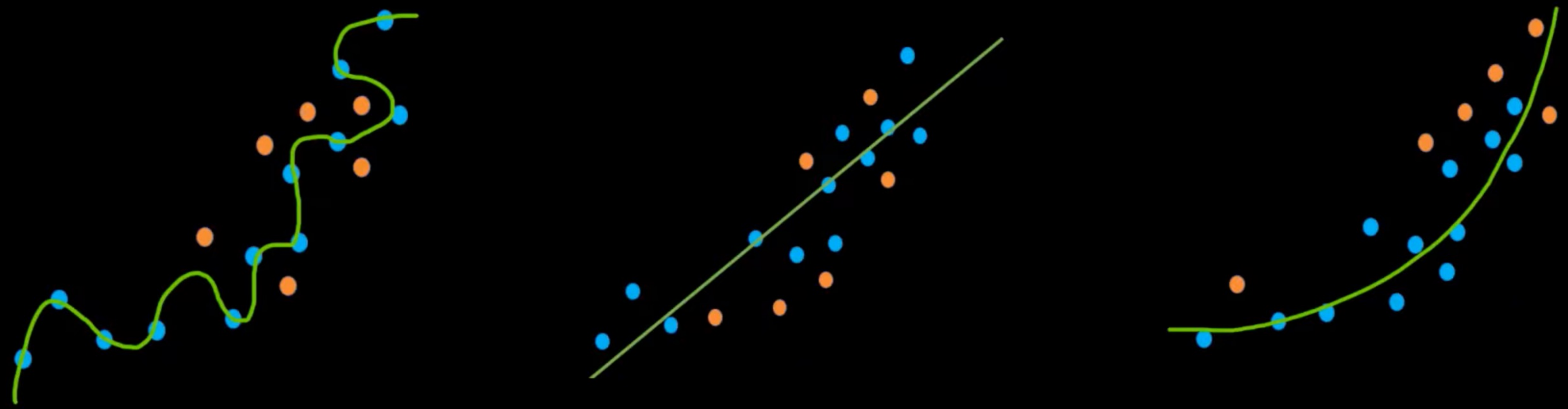
Aditya Koneru (akoner3@uic.edu)

Suvo Banik (sbanik2@uic.edu)

Learning objectives for this class

- Concept of training vs. test dataset
- Understand the difference between bias and variance
- Understand the concept of underfitting, overfitting and balanced fit
- Difference between validation and test dataset
- Ways to reduce overfitting
- K-fold and leave one out cross-validation

Overfitting vs. Underfitting vs. Balanced Fit

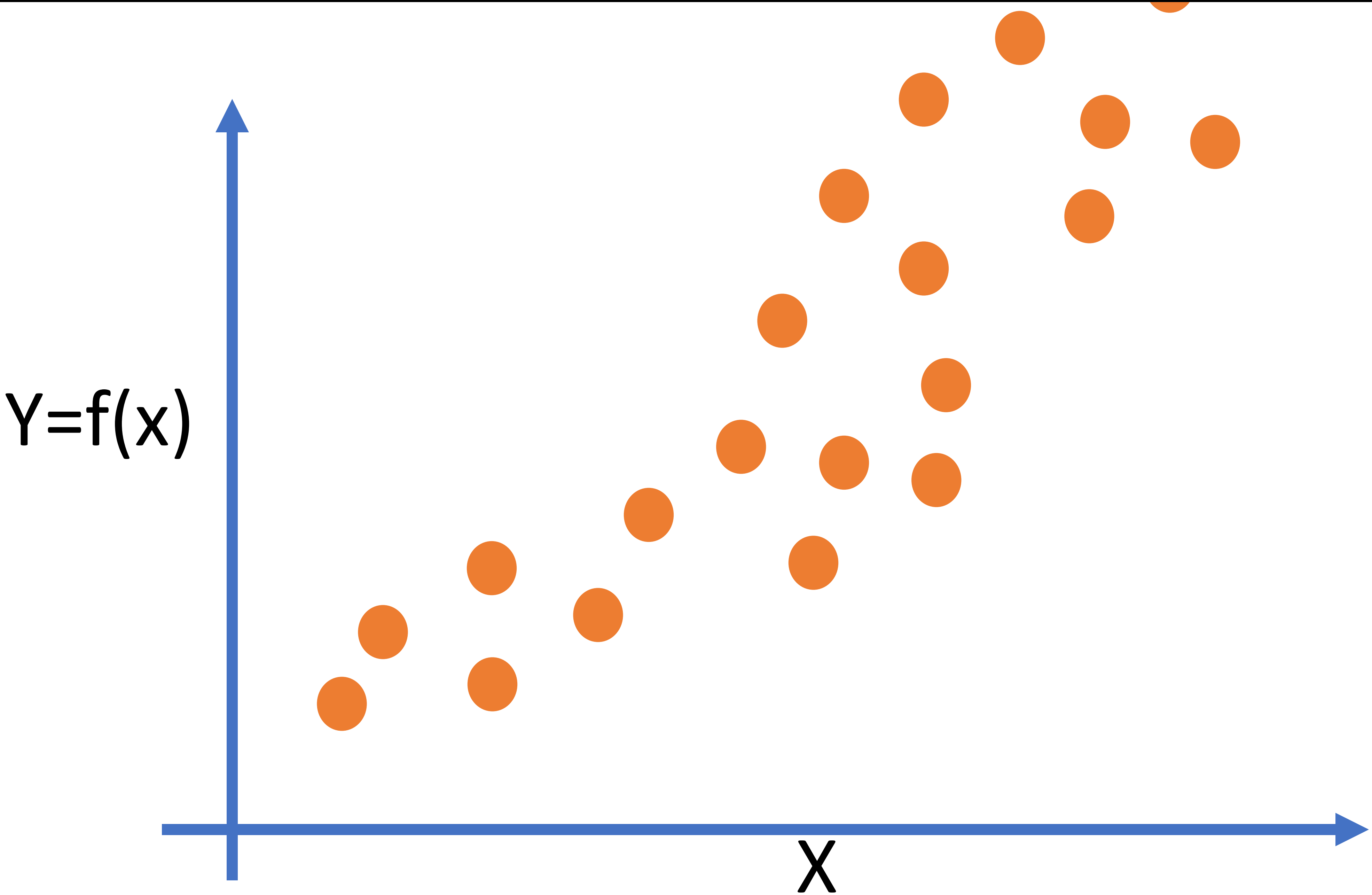


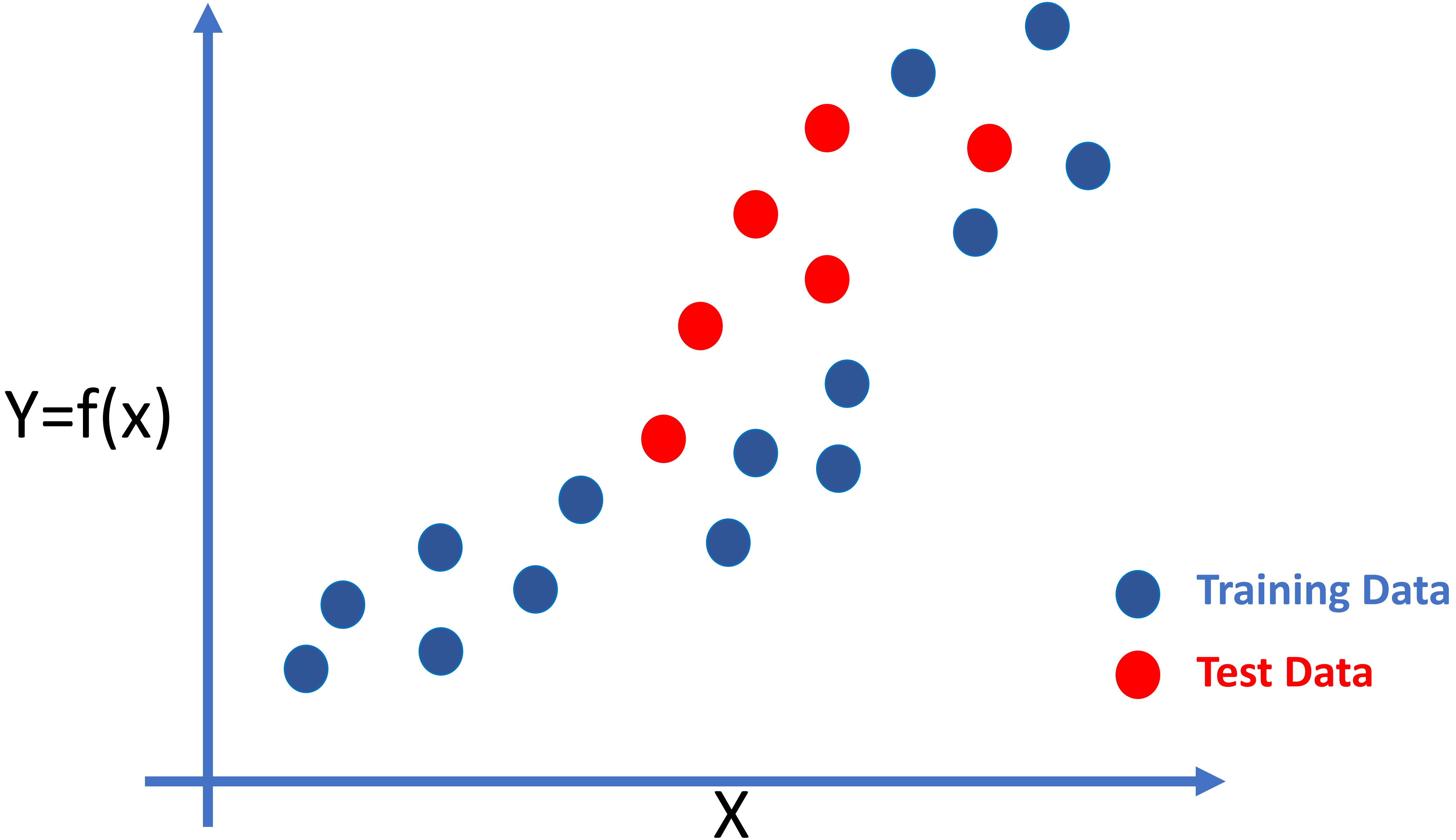
overfit

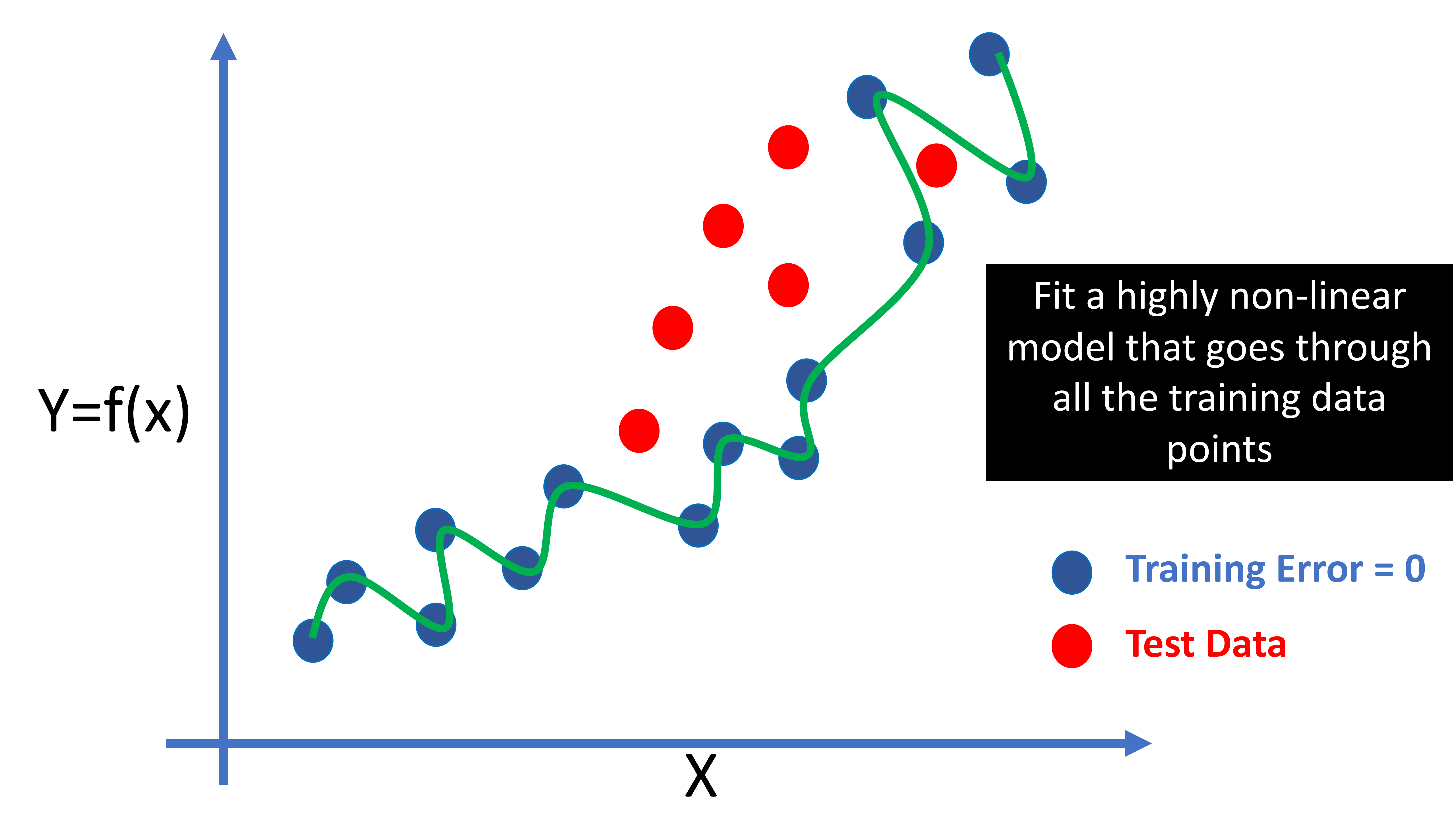
underfit

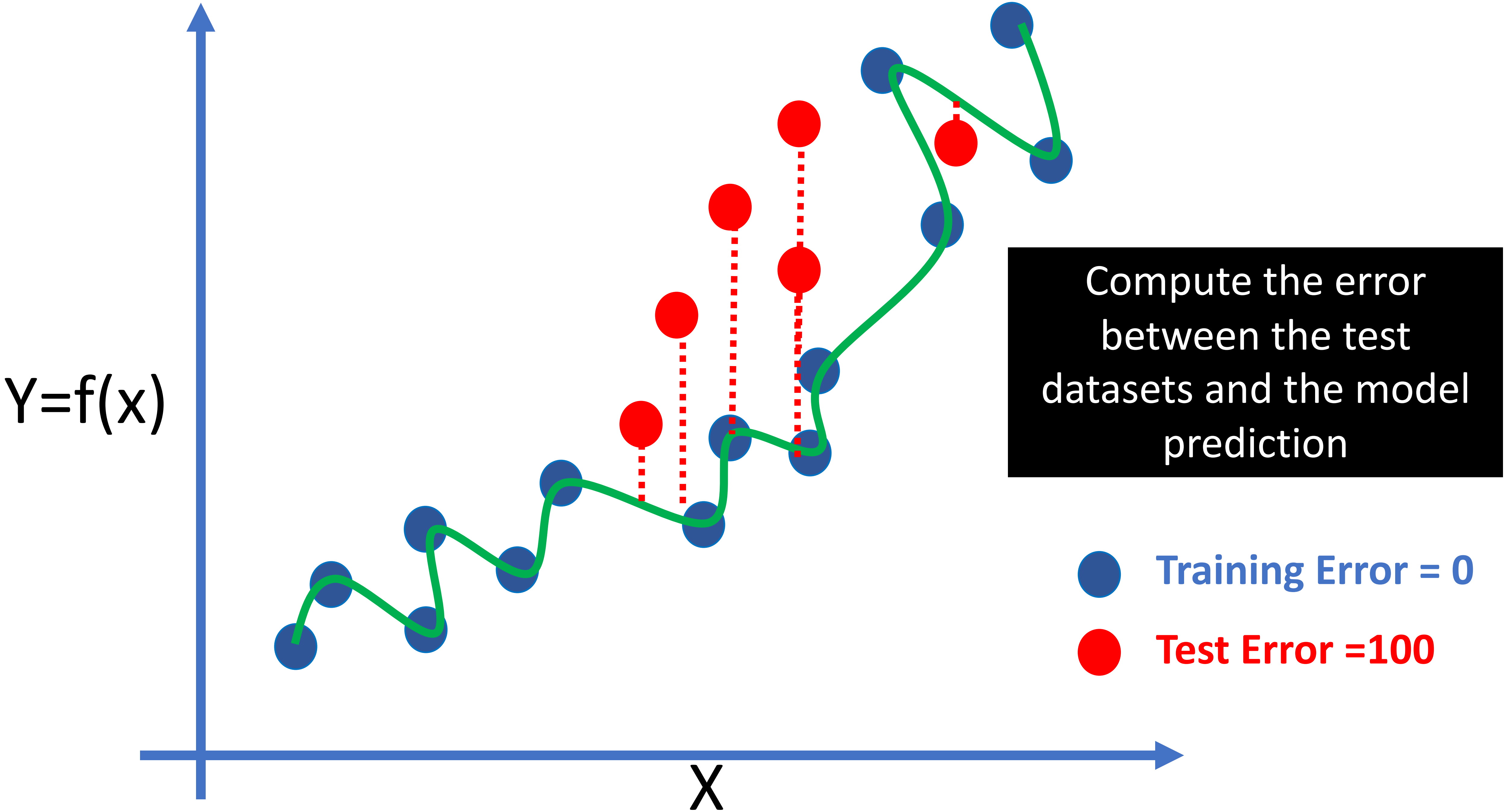
balanced fit

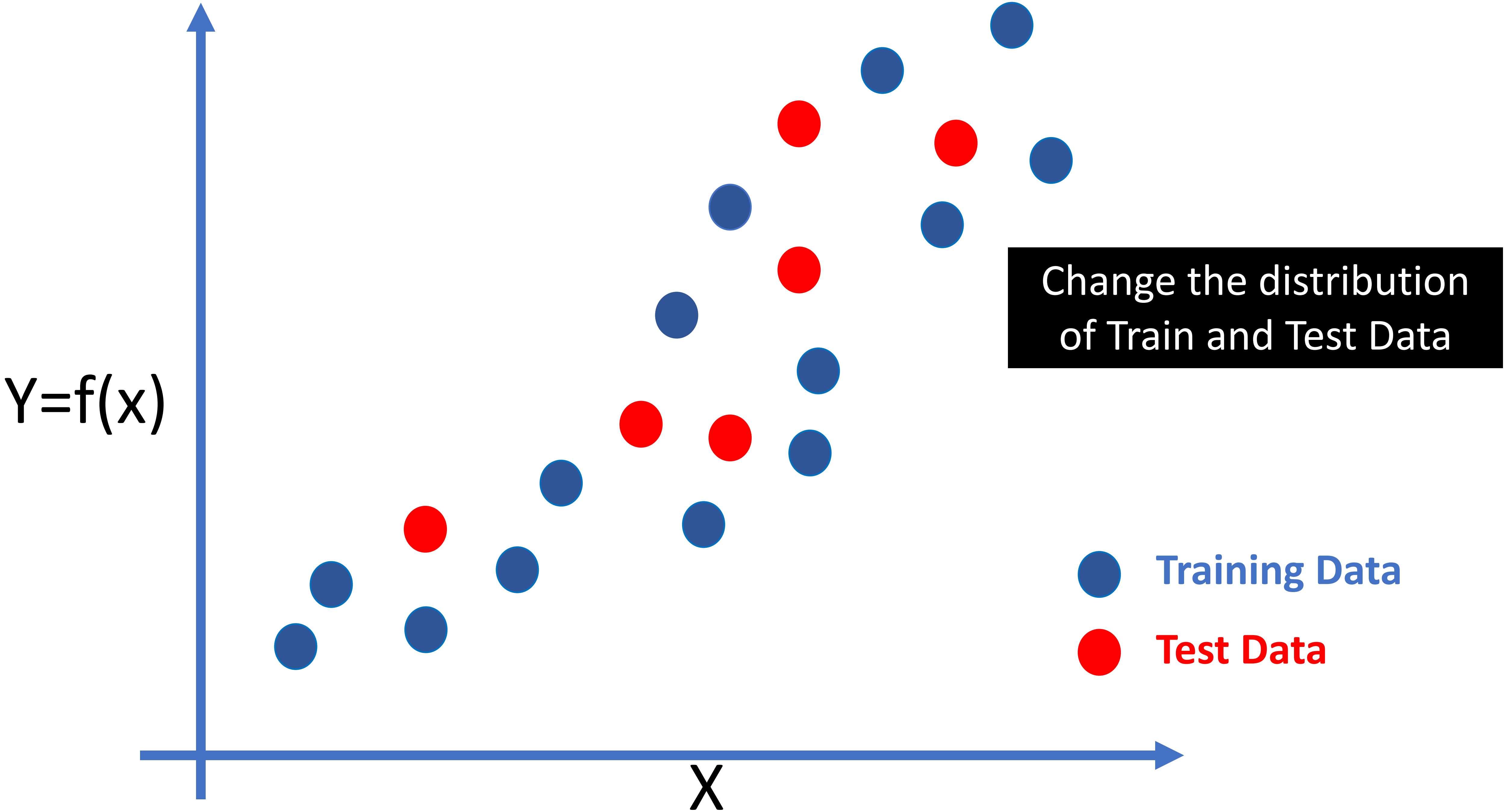
How do we select train and test dataset?

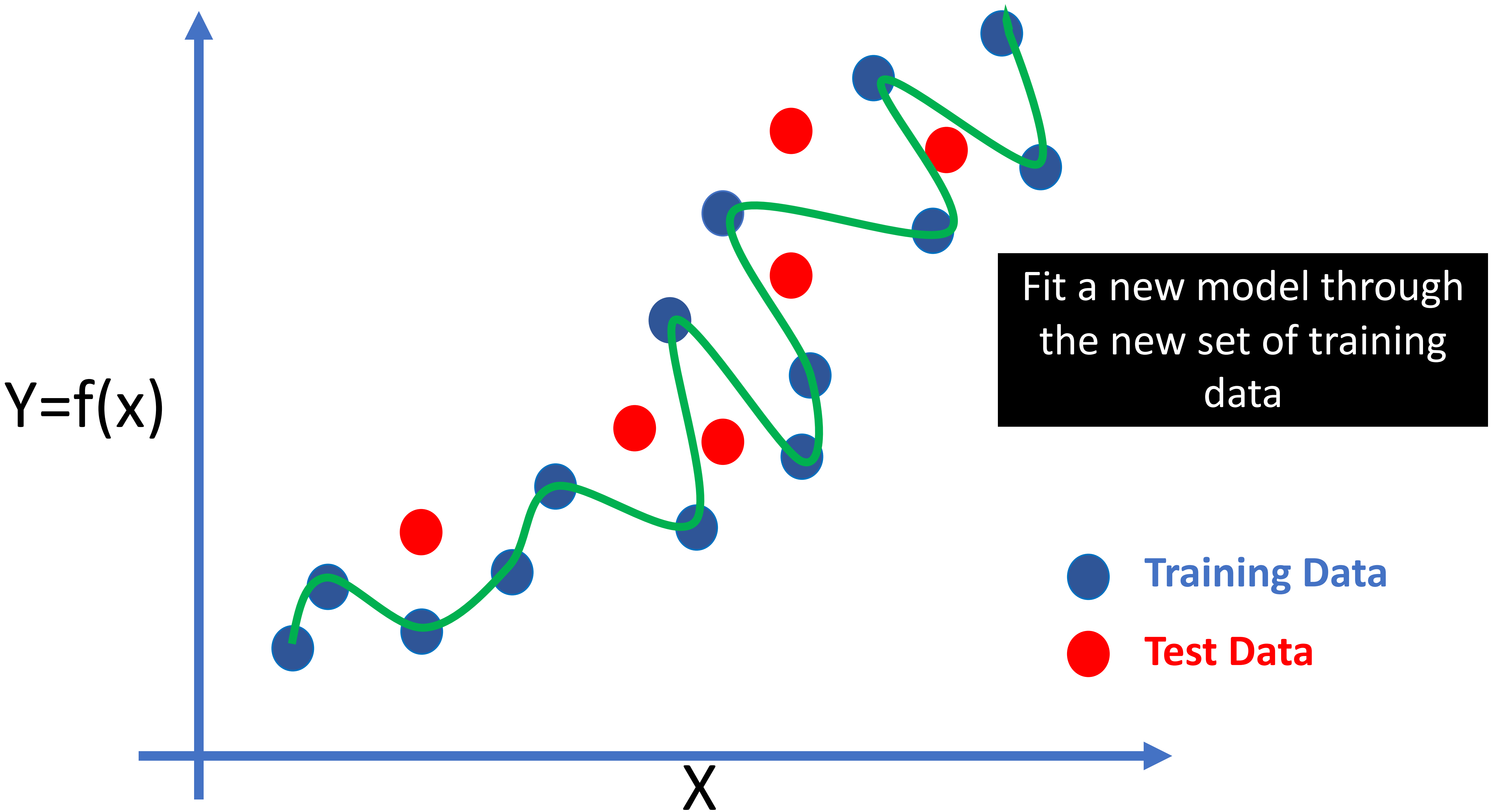


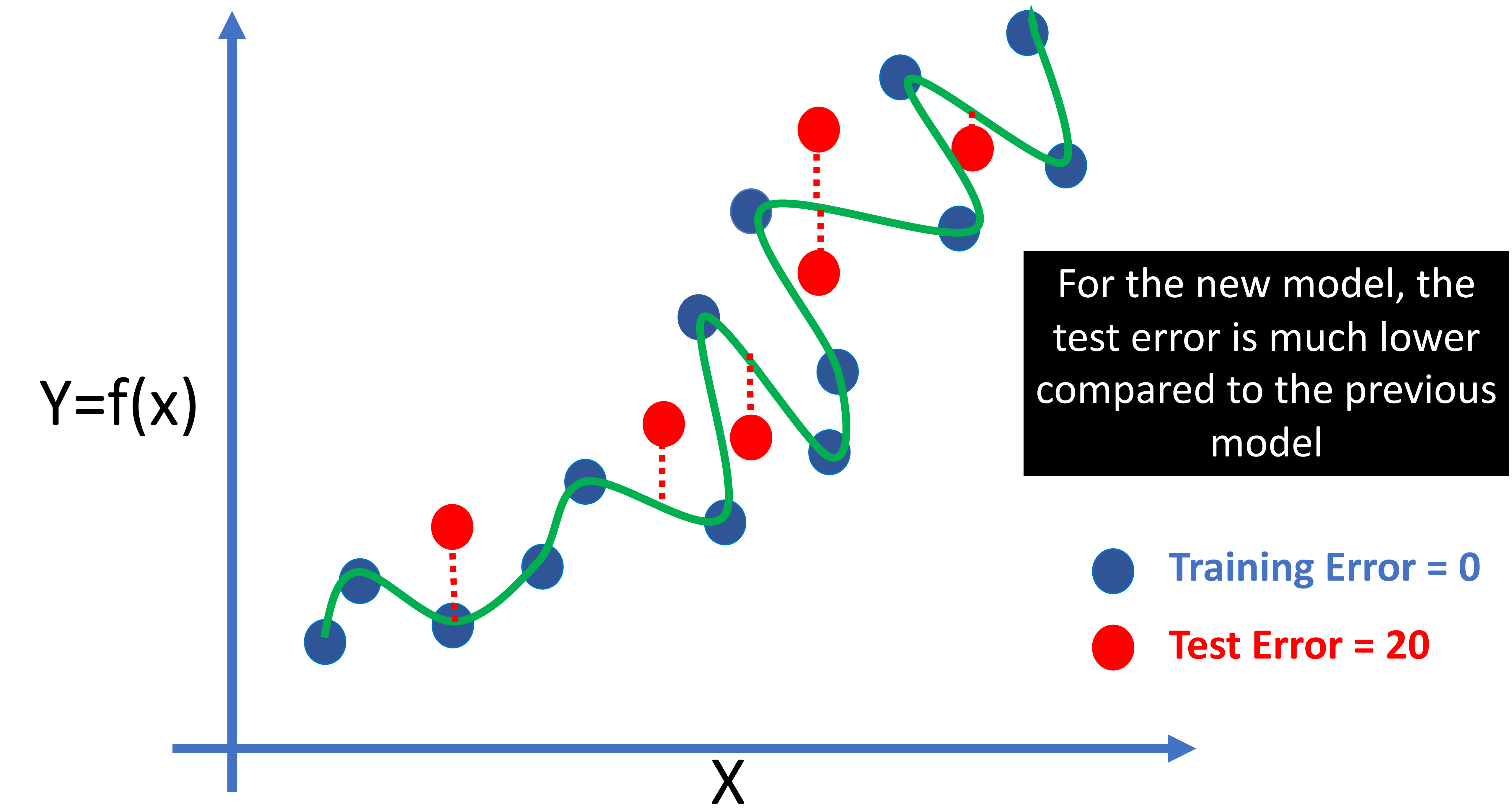

Y=f(x)
X
Training Data
Test Data


Y=f(x)
X
Fit a highly non-linear model that goes through all the training data points
Training Error = 0
Test Data


Y=f(x)
X
Compute the error between the test datasets and the model prediction
Training Error = 0
Test Error =100


Change the distribution of Train and Test Data
Y=f(x)
X
Training Data
Test Data


Y=f(x)
X
Fit a new model through the new set of training data
Training Data
Test Data


Y=f(x)
X
For the new model, the test error is much lower compared to the previous model
Training Error = 0
Test Error = 20

Example of High Variance

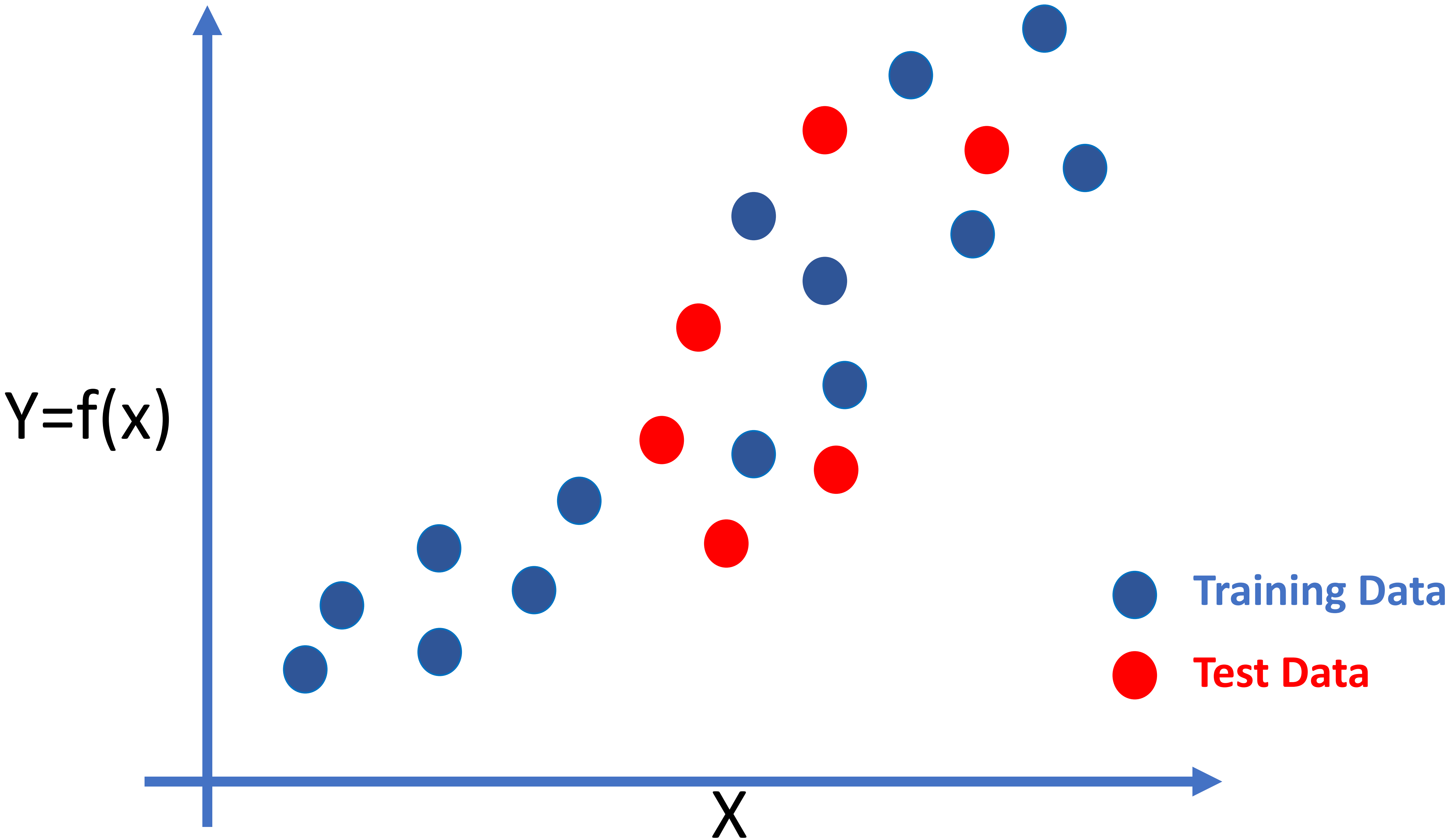
TEST ERROR =20

TEST ERROR =100

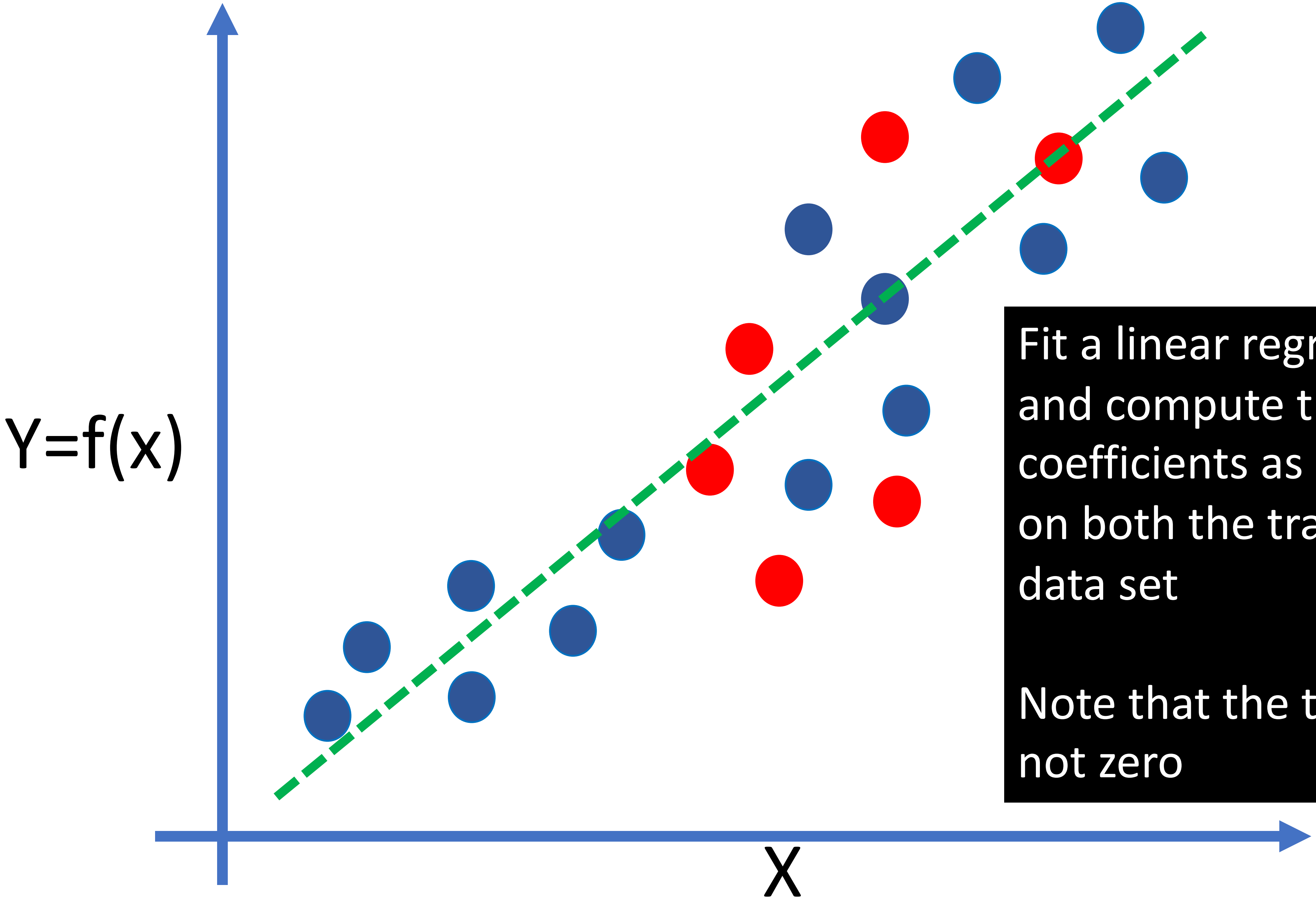
Example of High Variance

This is a case of High Variance, where the Test Error varies greatly depending on the selection of the training dataset

Now what do mean by bias?

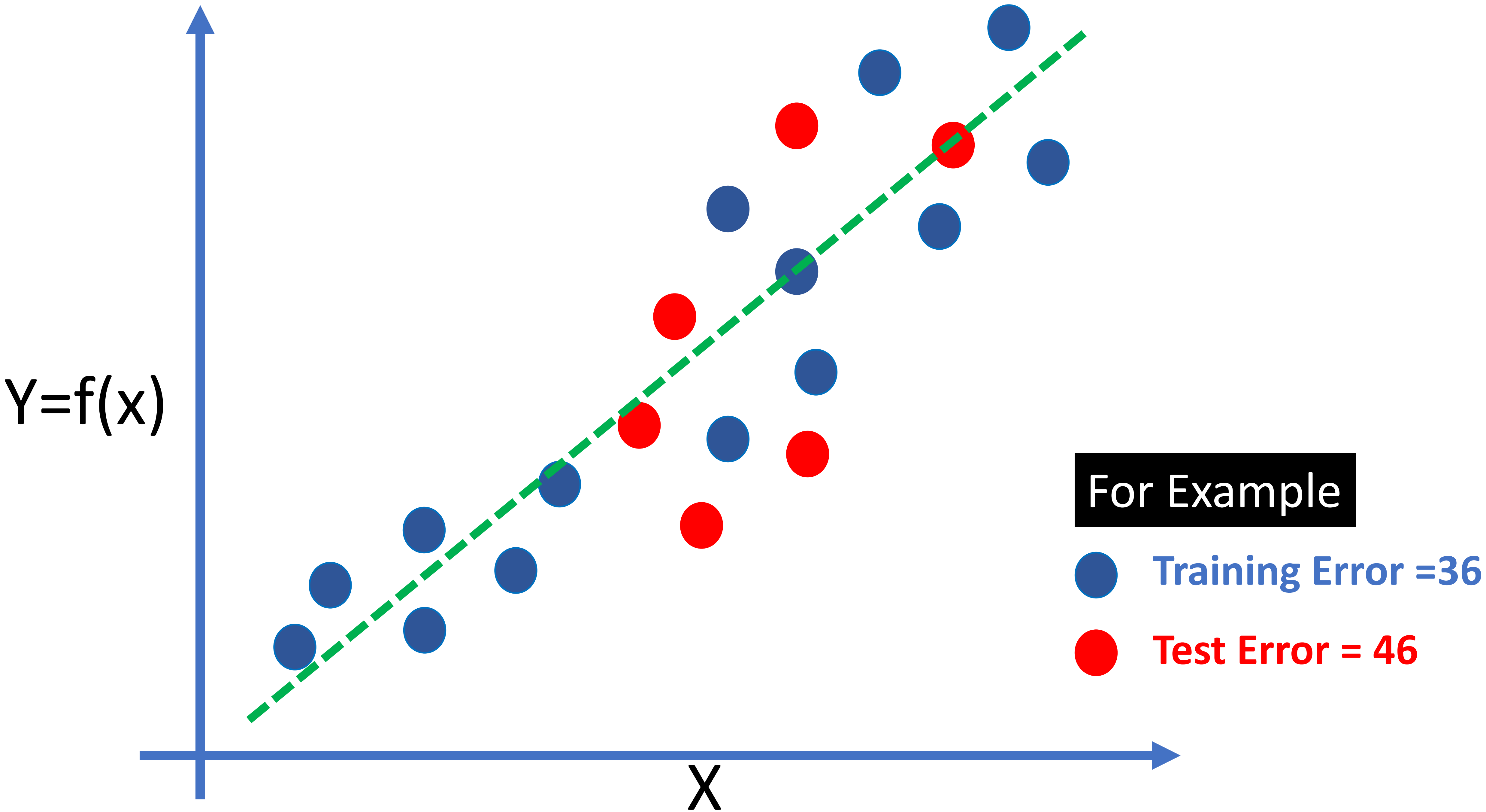


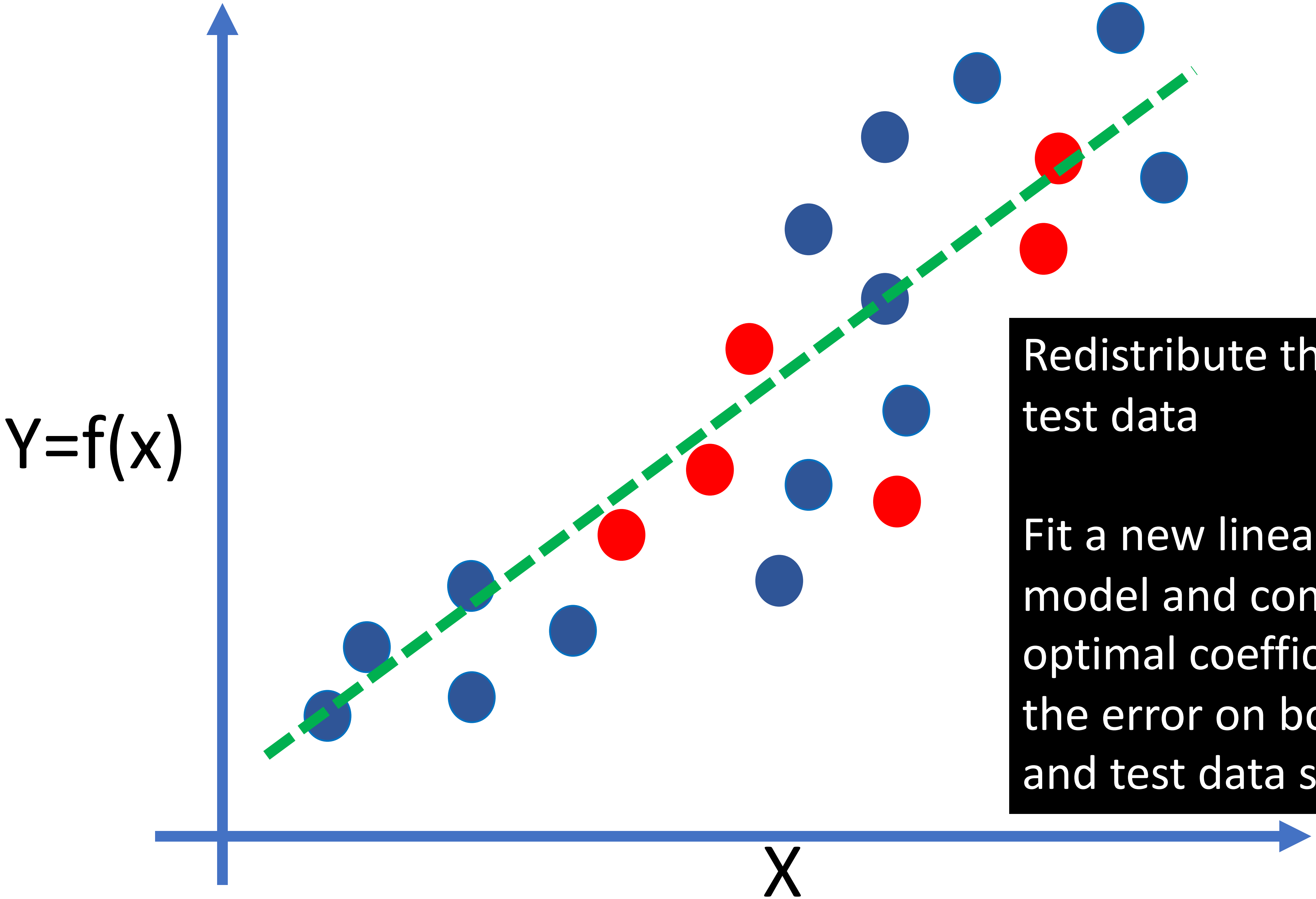
Y=f(x)
X
Training Data
Test Data



Fit a linear regression model and compute the optimal coefficients as well as the error on both the training and test data set

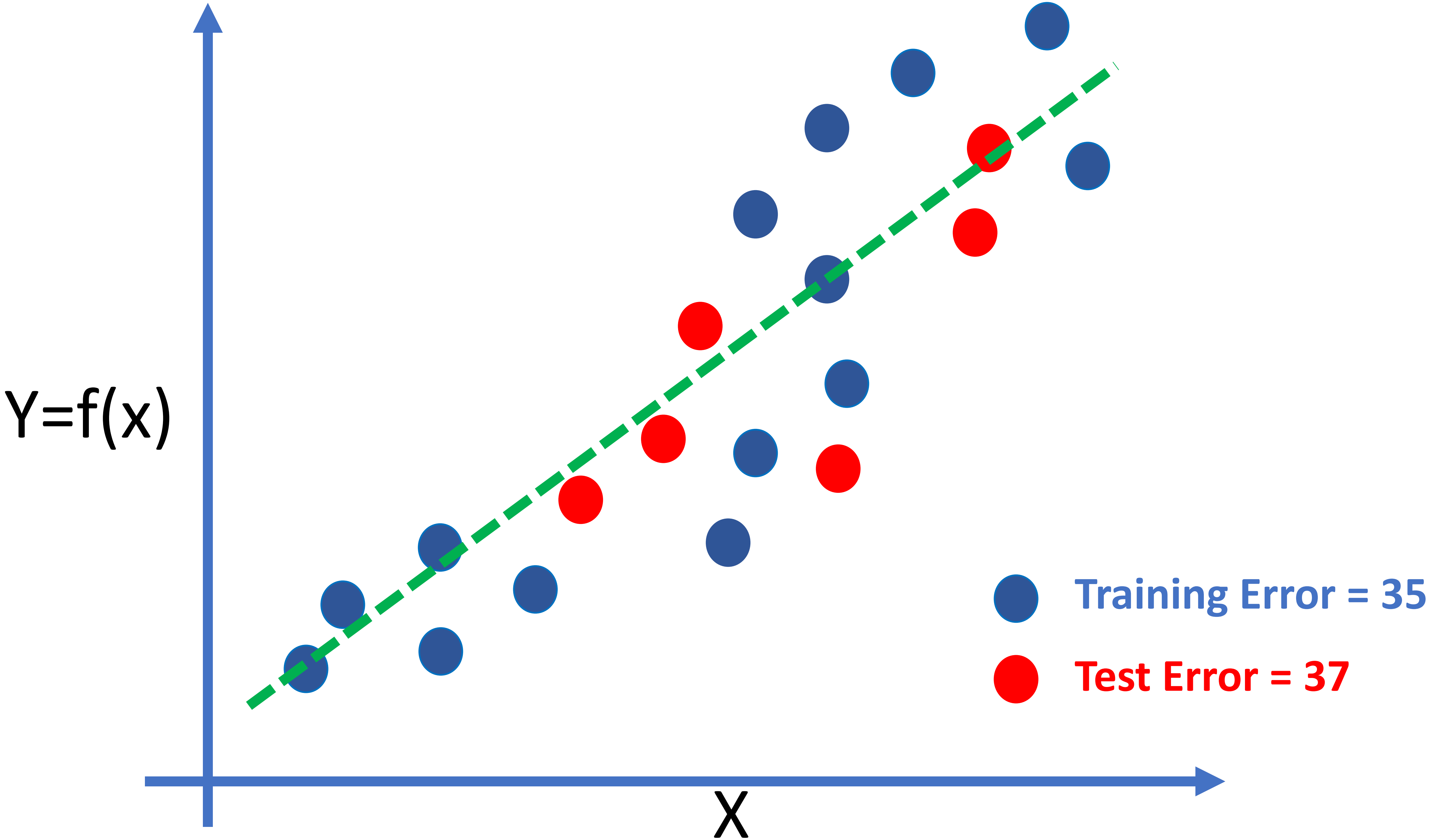
Note that the training error is not zero

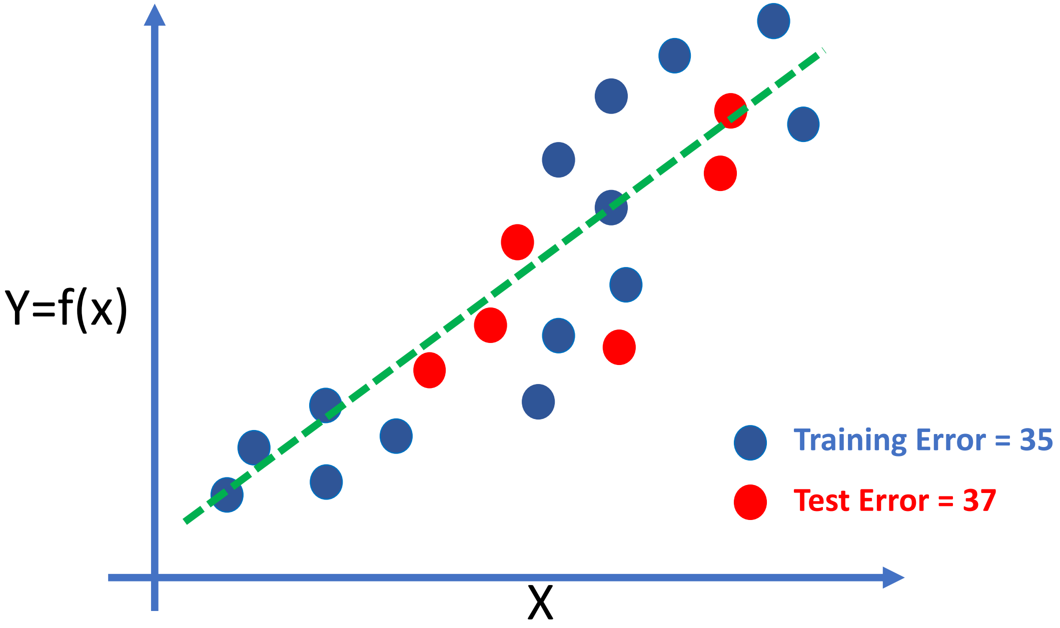

Y=f(x)
X
For Example
Training Error =36
Test Error = 46

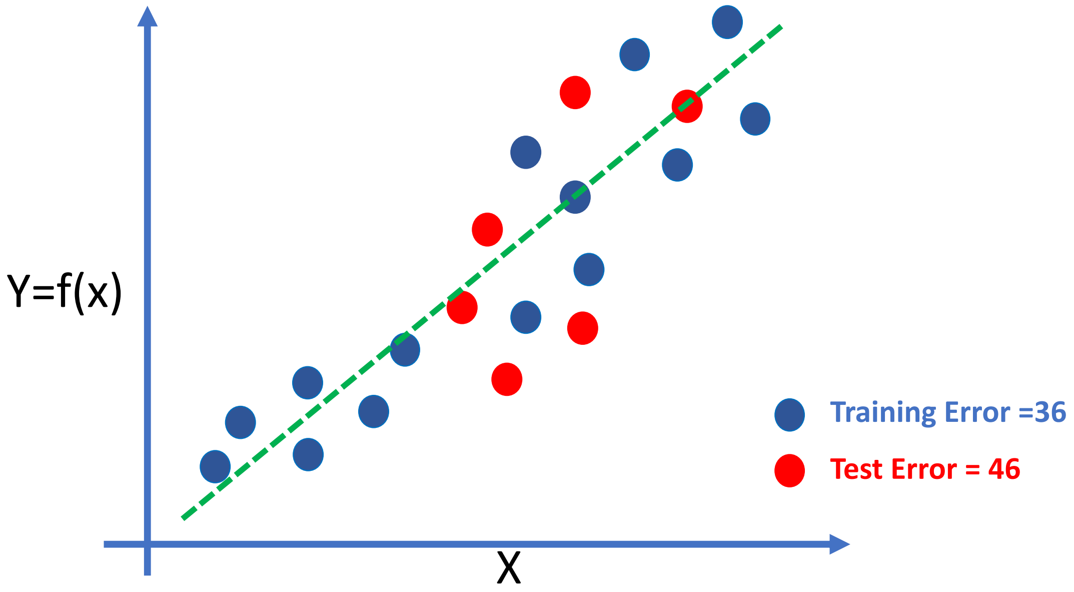


Redistribute the training and test data

Fit a new linear regression model and compute the optimal coefficients as well as the error on both the training and test data set


Y=f(x)
X
Training Error = 35
Test Error = 37



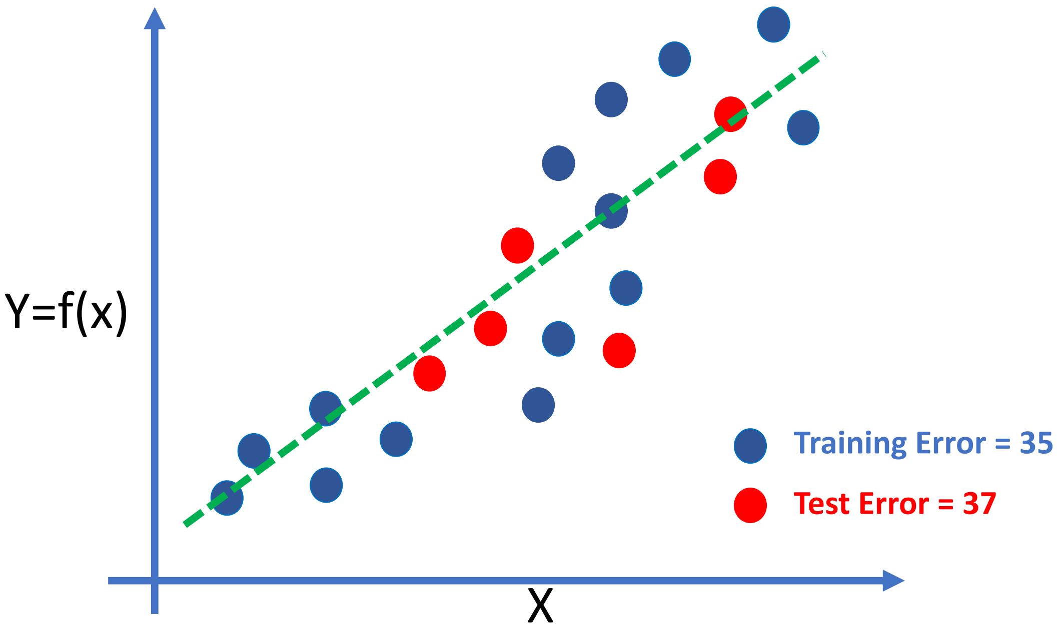


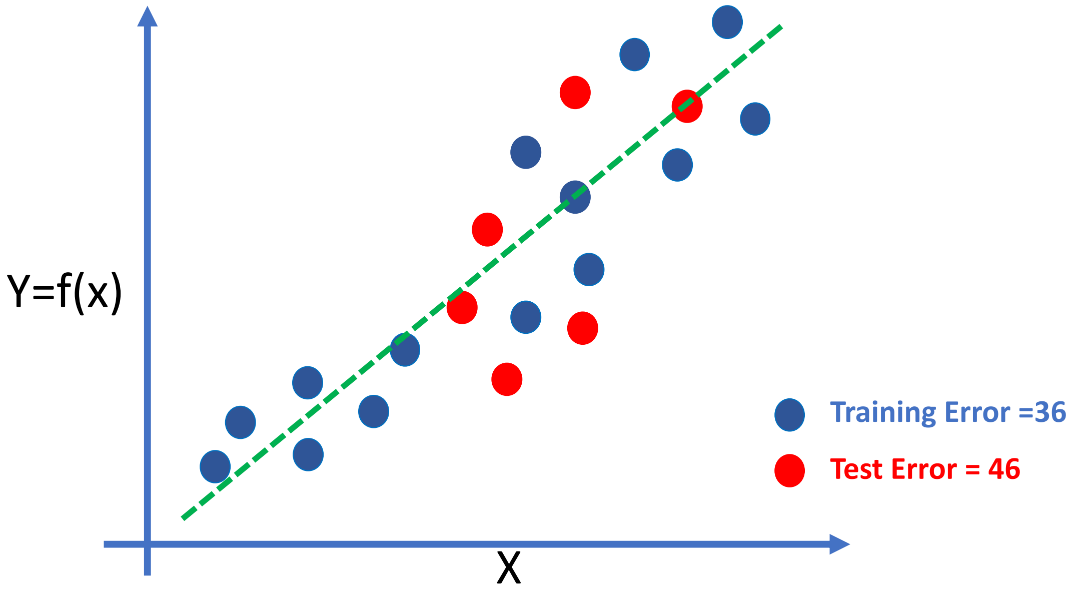
Train ERROR = 35

High Bias

Train ERROR = 36

Bias is a measurement of how accurately a model can capture a pattern in a training dataset. In above case since train error is big it is said to have a high bias.

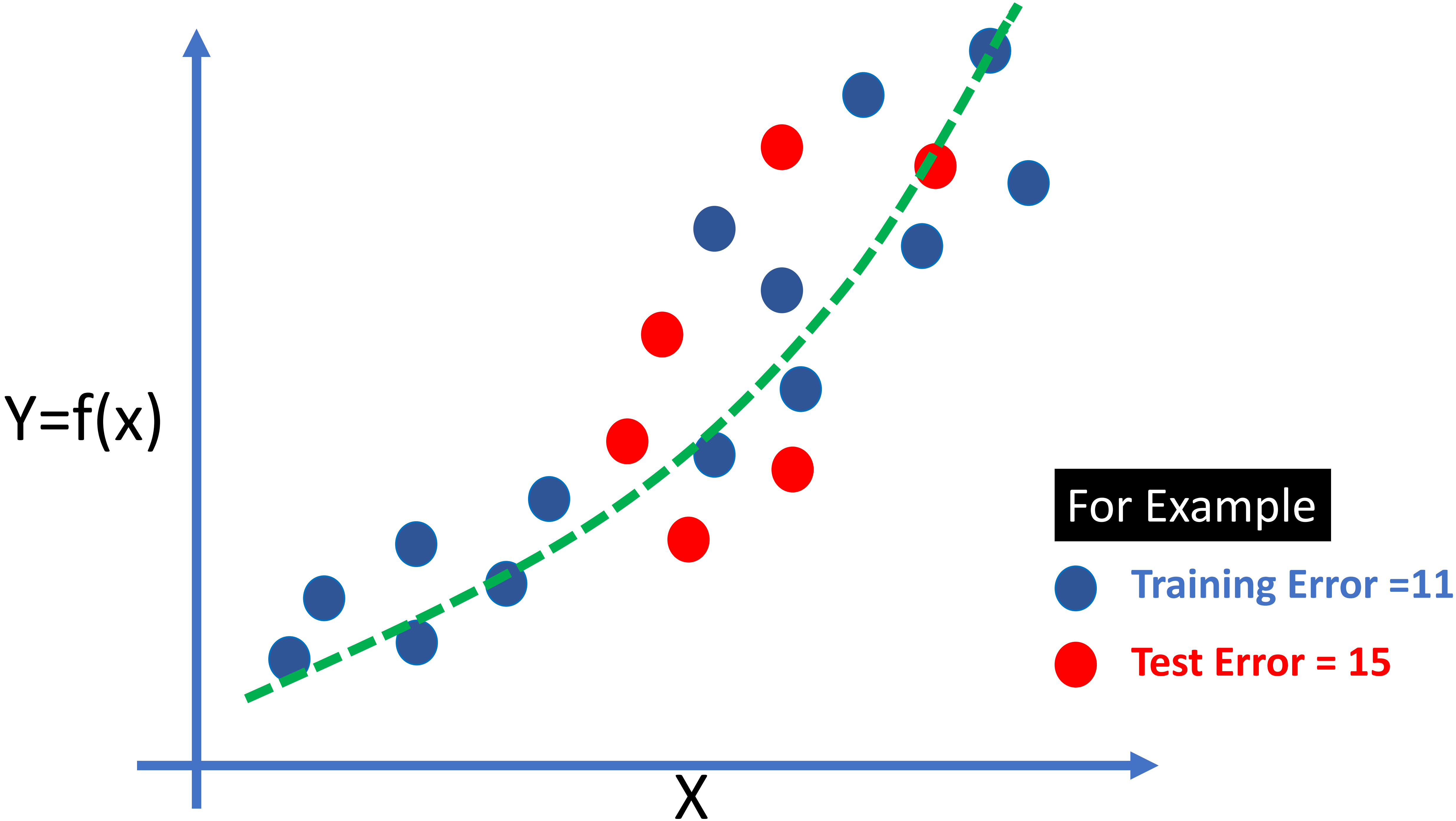


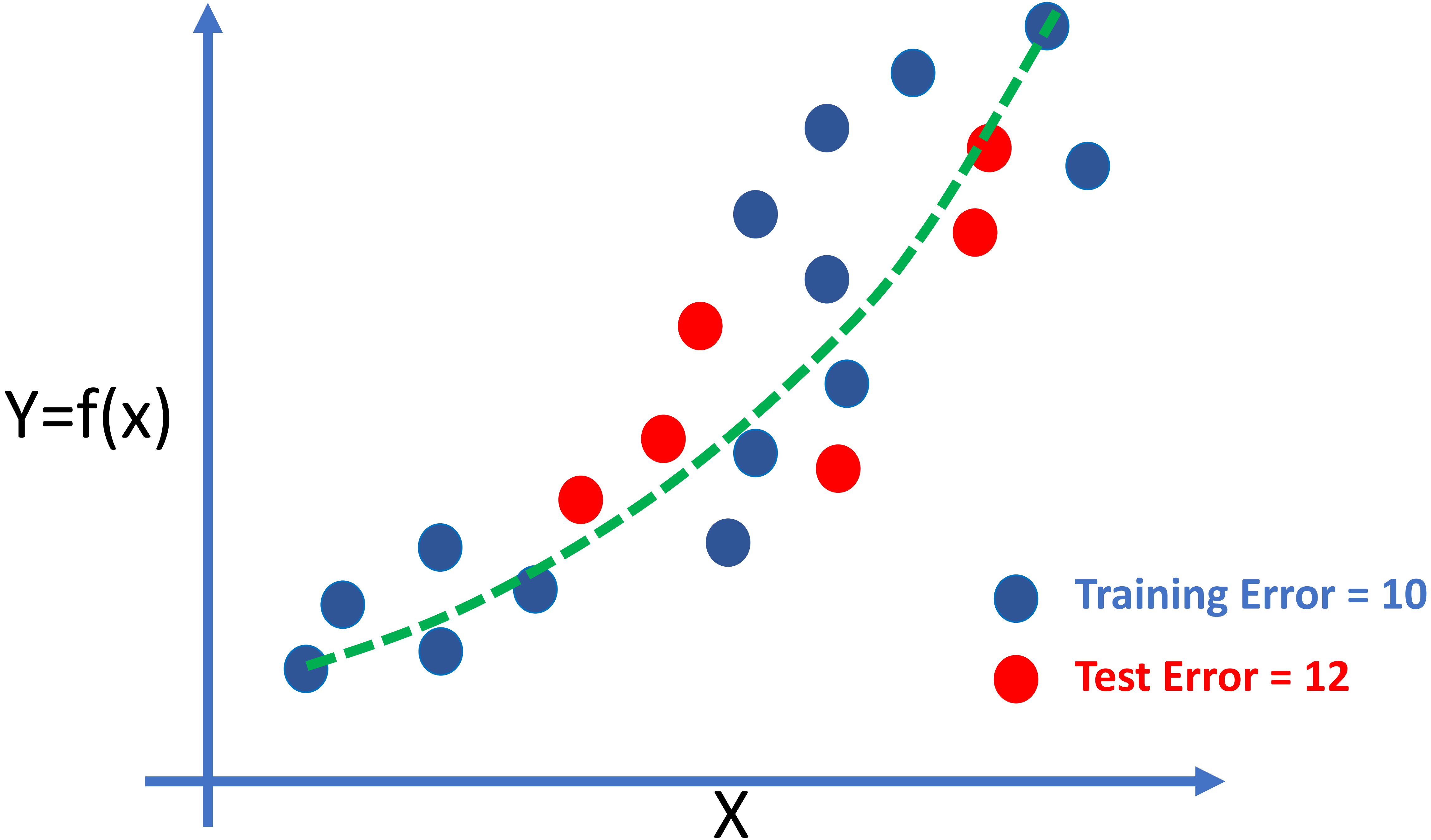


TEST ERROR = 37

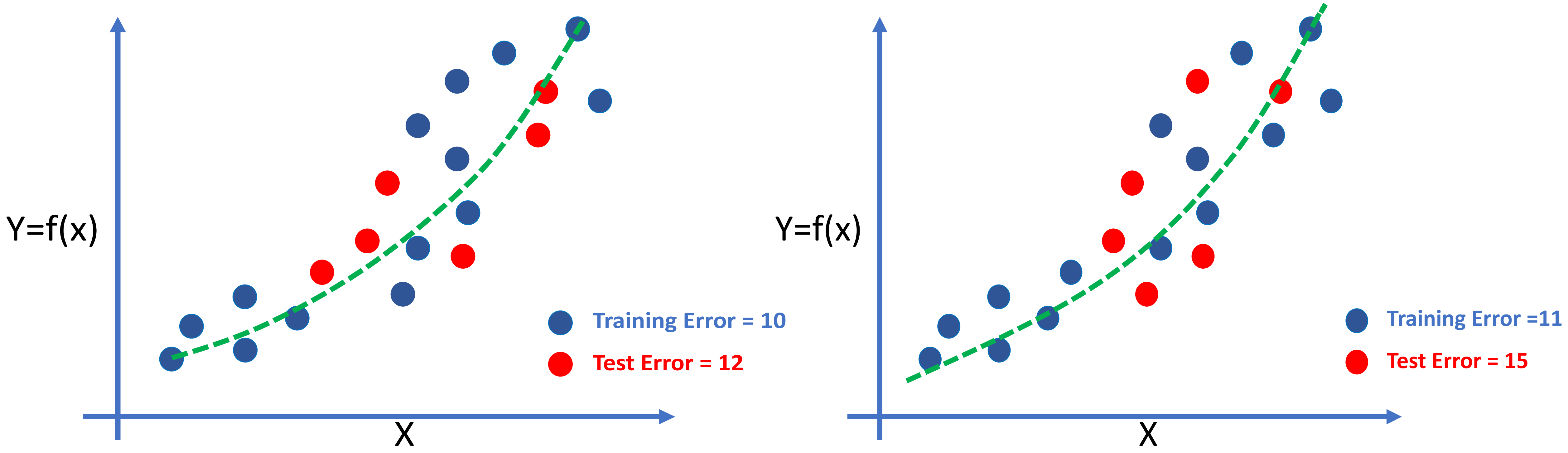
Low Variance

TEST ERROR = 46


Y=f(x)
X
For Example
Training Error =11
Test Error = 15

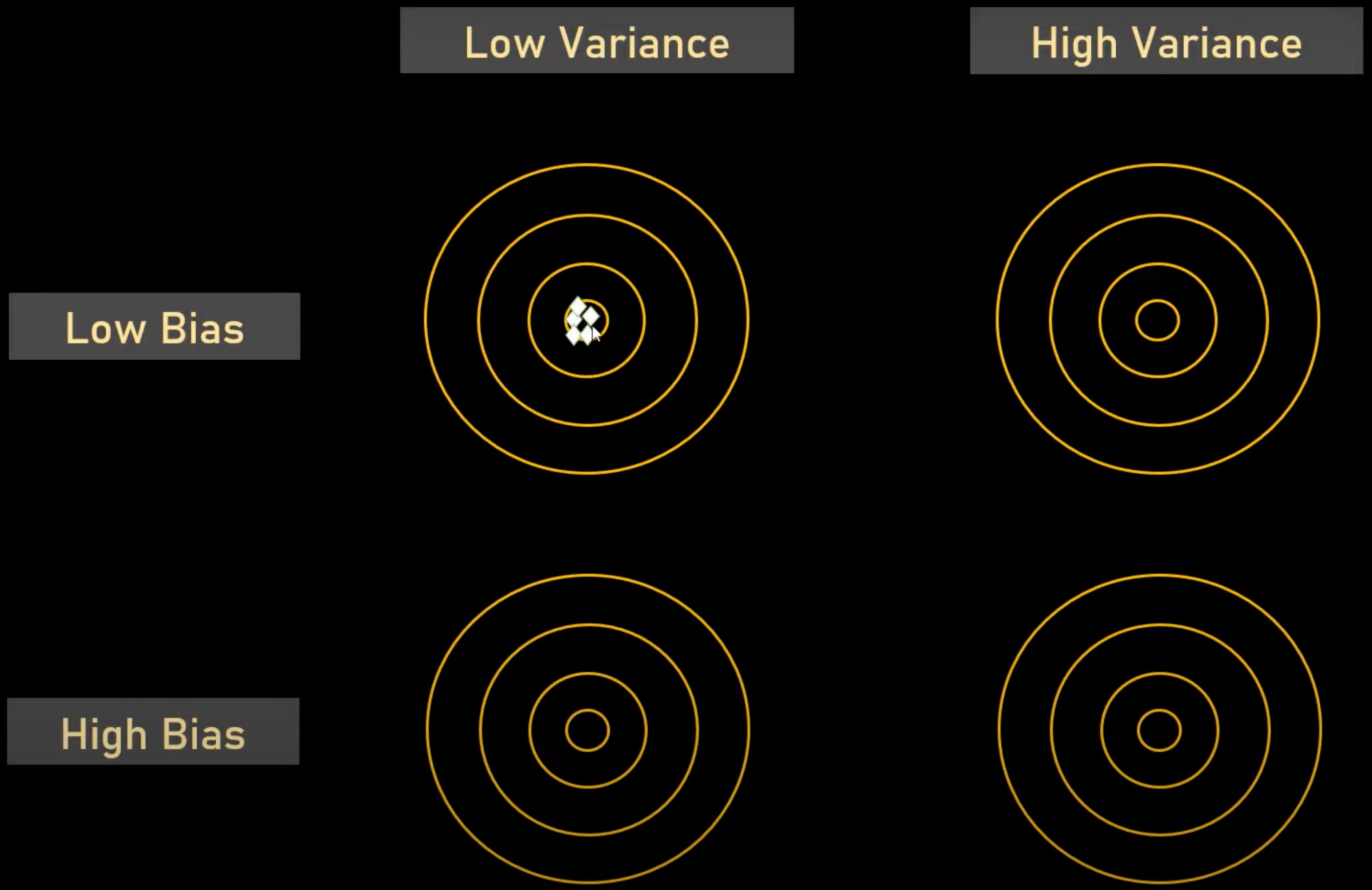

Y=f(x)
X
Training Error = 10
Test Error = 12

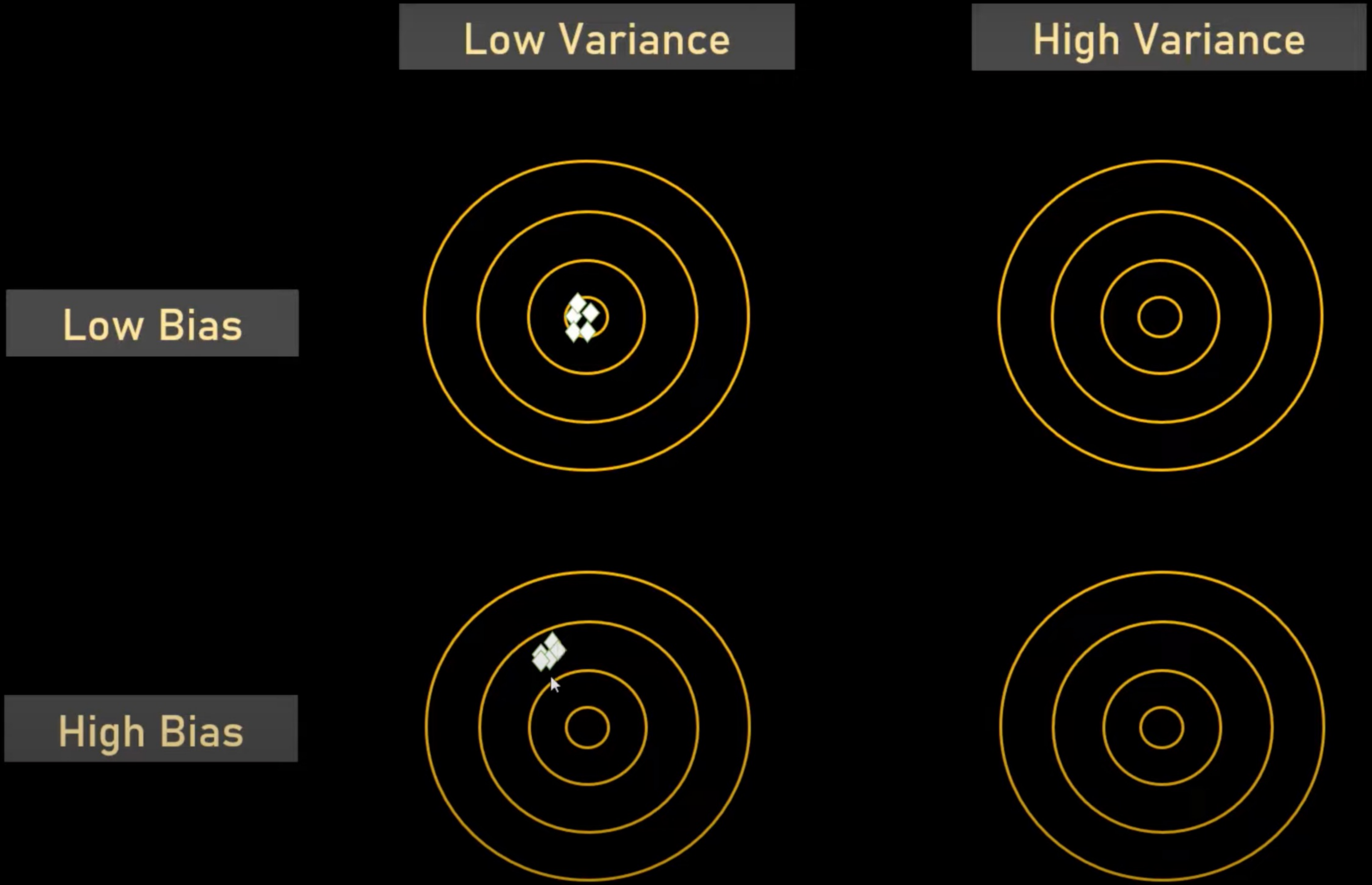
Balanced Fit

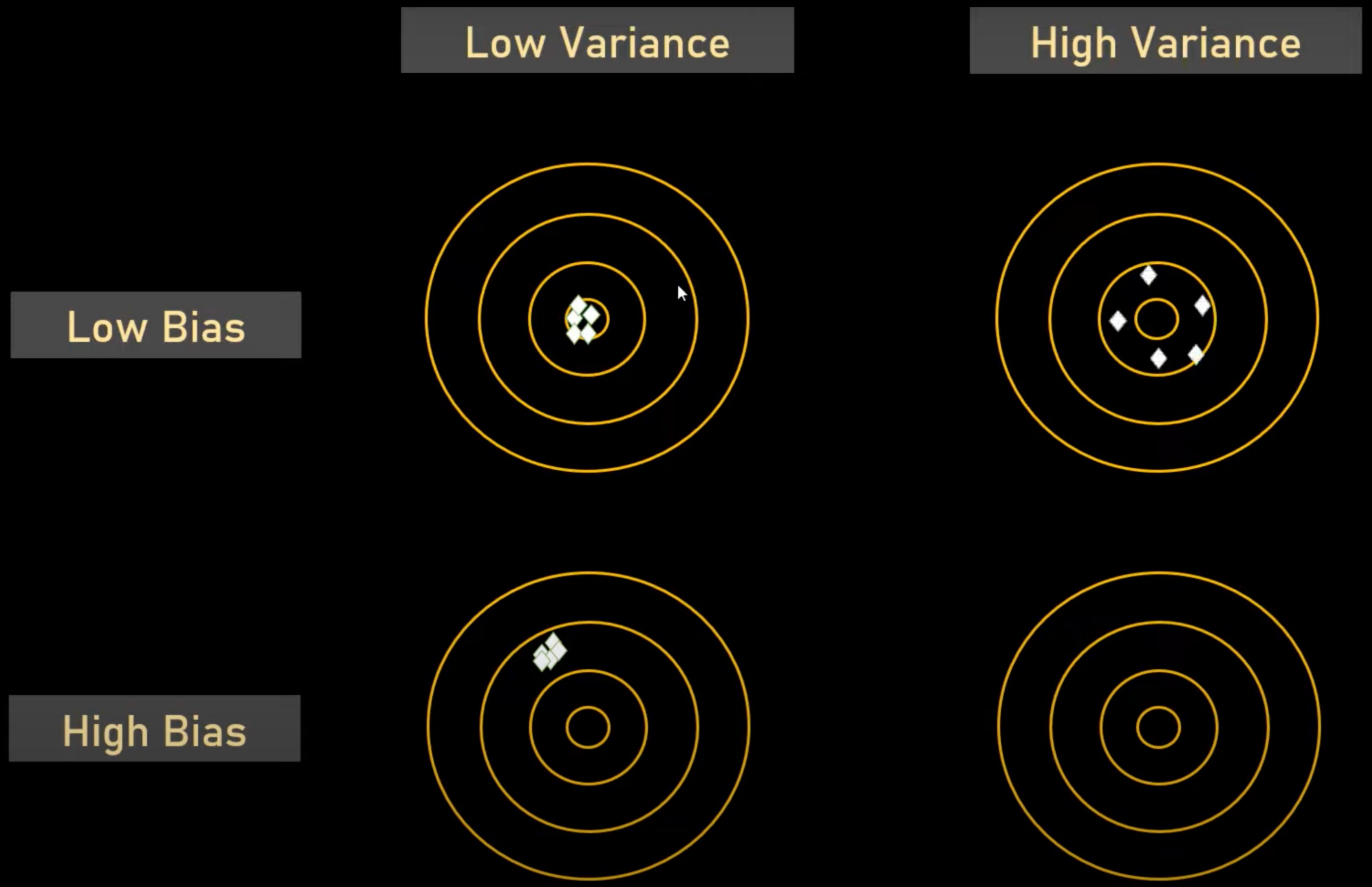


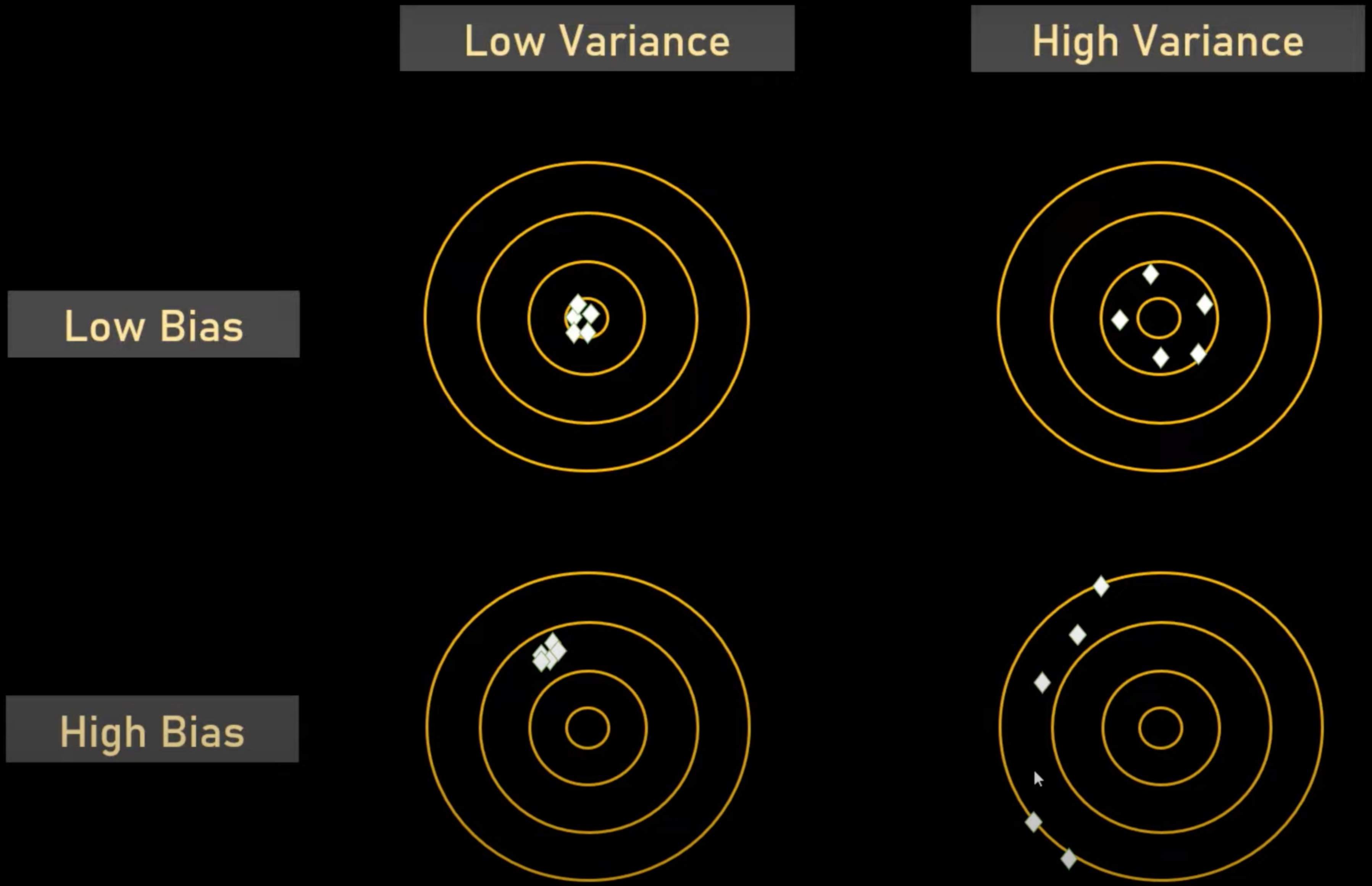
Test error = 12	low variance	Test error = 15
Train error = 10	low bias	Train error = 11

Nature of the fit depends on model complexity and data


Low Variance
High Variance
Low Bias
High Bias


Low Variance
High Variance
Low Bias
High Bias


Low Variance
High Variance
Low Bias
High Bias


Low Variance
High Variance
Low Bias
High Bias

Methods to reduce overfitting

- Hold-out (data)
- Cross-validation (data)
- Data augmentation (data)
- Feature selection or principal components (data)
- L1 / L2 regularization (learning algorithm)
- Dropout (model)
- Early stopping (model)
- Ensembling (model)

Training vs. Validation vs. Test Data

Training data is the initial data used to train machine learning models. The training data is used to teach the algorithms how to make predictions or perform a task.

Validation data set is used during the training phase to provide an unbiased evaluation of the model's performance. The validation data set is used to compare the performances of different candidate classifiers and decide which one to take. The validation data set is also used to minimize overfitting

Test data set is used after the model has been fully trained to assess the model's performance on completely unseen data. The test data set is used to verify that the accuracy is sufficient

Cross-validation

We want to use the variables (C.co, C.cr, C.cu, C.fe, C.ni) ….

	ID	HV	C.al	C.co	C.cr	C.cu	C.fe	C.ni
0	25	170	0.056604	0.000000	0.188679	0.188679	0.188679	0.377358
1	60	380	0.200000	0.266667	0.000000	0.000000	0.266667	0.266667
2	155	775	0.400000	0.200000	0.200000	0.000000	0.200000	0.000000
3	88	486	0.208333	0.000000	0.208333	0.208333	0.208333	0.166667
4	2	118	0.024390	0.243902	0.243902	0.000000	0.243902	0.243902
...	...	...	...	...	...	...	...	...
115	59	371	0.166667	0.000000	0.555556	0.000000	0.000000	0.277778
116	104	537	0.166667	0.166667	0.166667	0.083333	0.250000	0.166667
117	116	558	0.264706	0.147059	0.147059	0.147059	0.147059	0.147059
118	154	768	0.400000	0.133333	0.066667	0.133333	0.200000	0.066667
119	123	584	0.222222	0.222222	0.000000	0.111111	0.222222	0.222222

Cross-validation

to predict the hardness of a material

	ID	HV	C.al	C.co	C.cr	C.cu	C.fe	C.ni
0	25	170	0.056604	0.000000	0.188679	0.188679	0.188679	0.377358
1	60	380	0.200000	0.266667	0.000000	0.000000	0.266667	0.266667
2	155	775	0.400000	0.200000	0.200000	0.000000	0.200000	0.000000
3	88	486	0.208333	0.000000	0.208333	0.208333	0.208333	0.166667
4	2	118	0.024390	0.243902	0.243902	0.000000	0.243902	0.243902
...	..	...	...	...	...	...	...	...
115	59	371	0.166667	0.000000	0.555556	0.000000	0.000000	0.277778
116	104	537	0.166667	0.166667	0.166667	0.083333	0.250000	0.166667
117	116	558	0.264706	0.147059	0.147059	0.147059	0.147059	0.147059
118	154	768	0.400000	0.133333	0.066667	0.133333	0.200000	0.066667
119	123	584	0.222222	0.222222	0.000000	0.111111	0.222222	0.222222

We can measure these variables

HV	C.al	C.co	C.cr	C.cu	C.fe	C.ni
????						

And predict the hardness of the new data

Other than overfitting, you will often have to decide which ML model is the best

Cross-validation

Imagine that this block represents the entire dataset

We need to perform at least two tasks with this data:

1) Estimate the coefficients/weights/parameters of the model (aka Training)

2) Evaluate the performance of the model in making predictions (aka Testing)

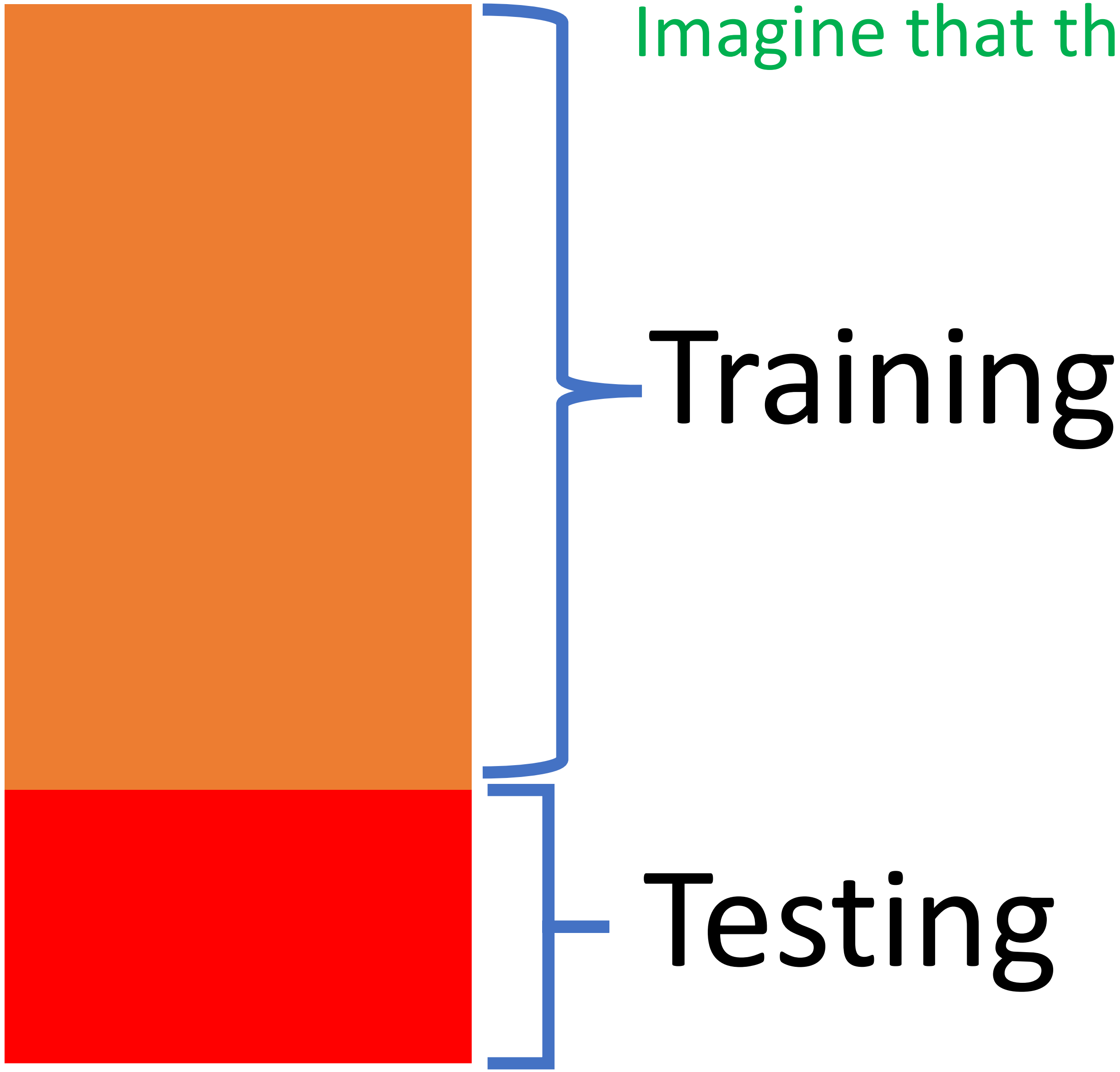
Cross-validation

Imagine that this block represents the entire dataset

Reusing all the data for training and testing is a bad idea as we won’t know if the predictions will be good on the data it was not trained on

Cross-validation

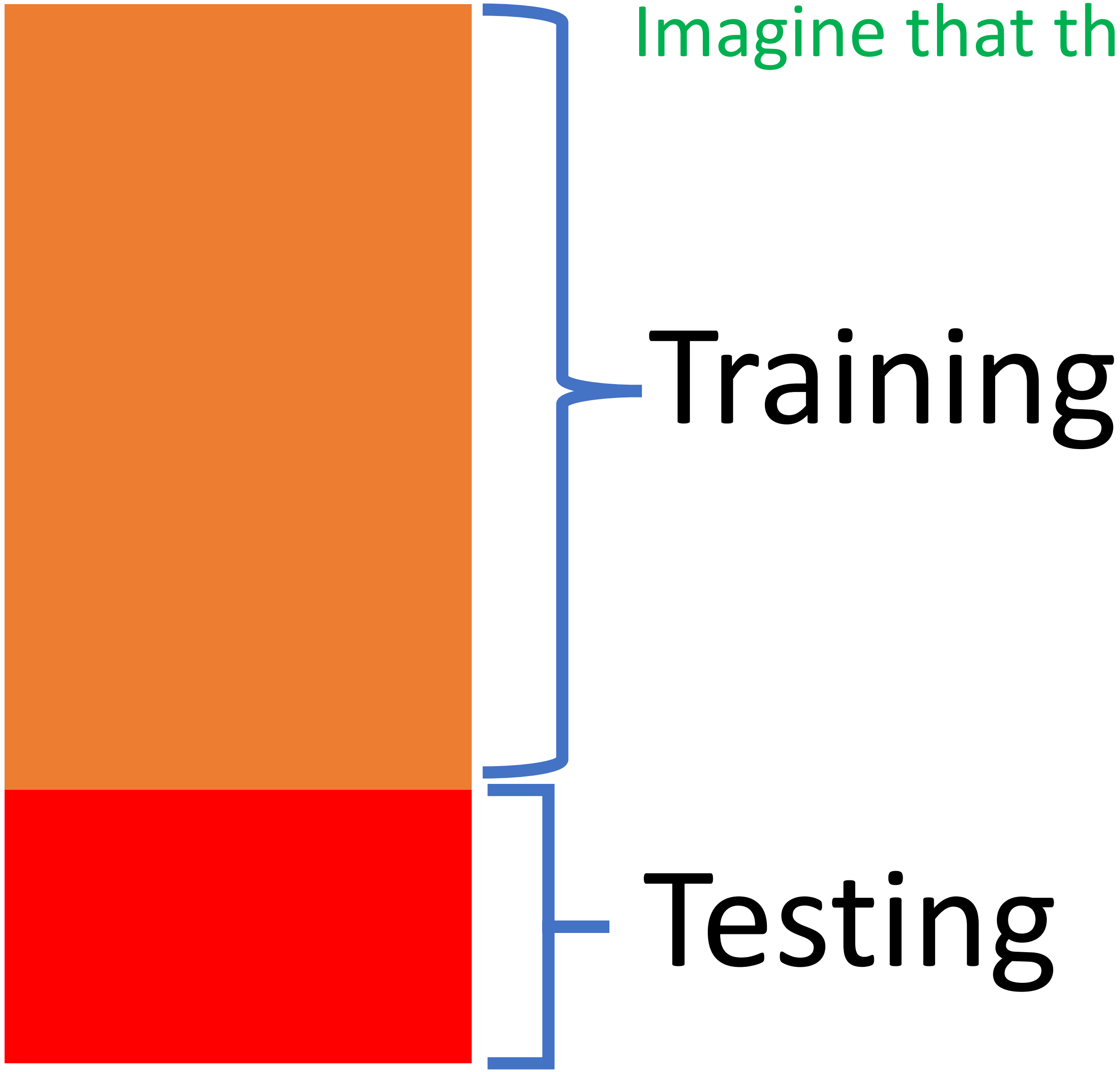
Imagine that this block represents the entire dataset



Typically, we divide the dataset into 80% for training and the remaining 20% for testing

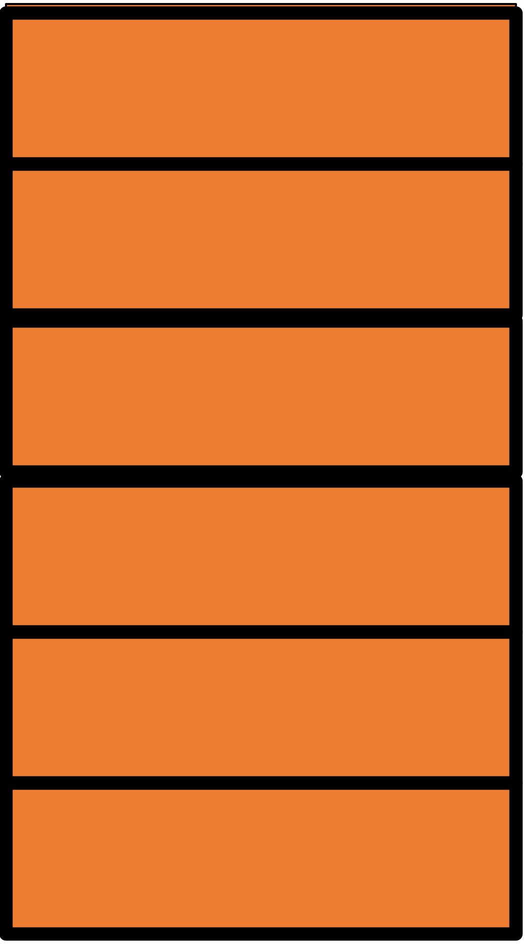
Cross-validation

Imagine that this block represents the entire dataset



Typically, we divide the dataset into 80% for training and the remaining 20% for testing

Cross-validation

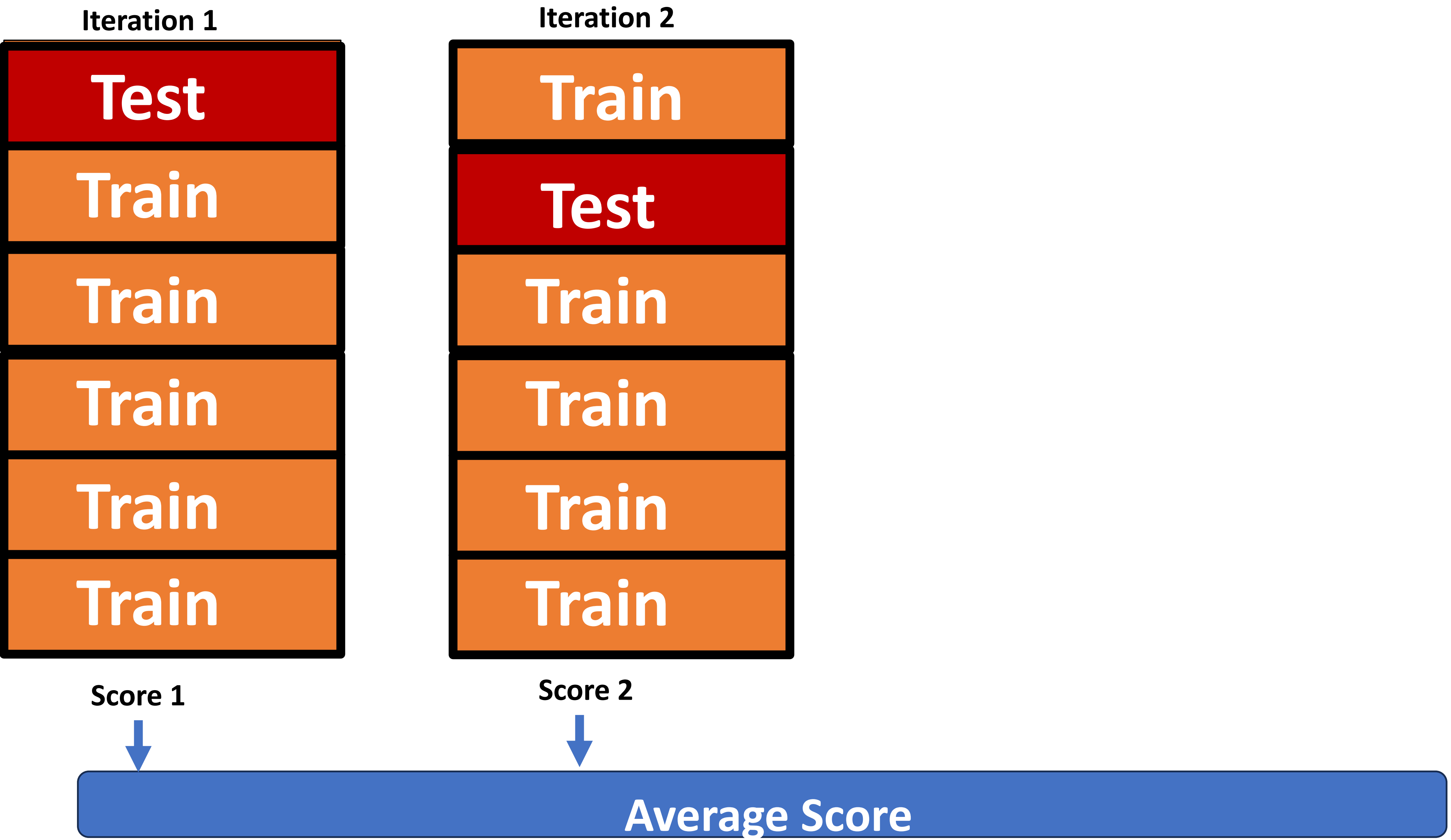


Divide the training dataset into k-blocks

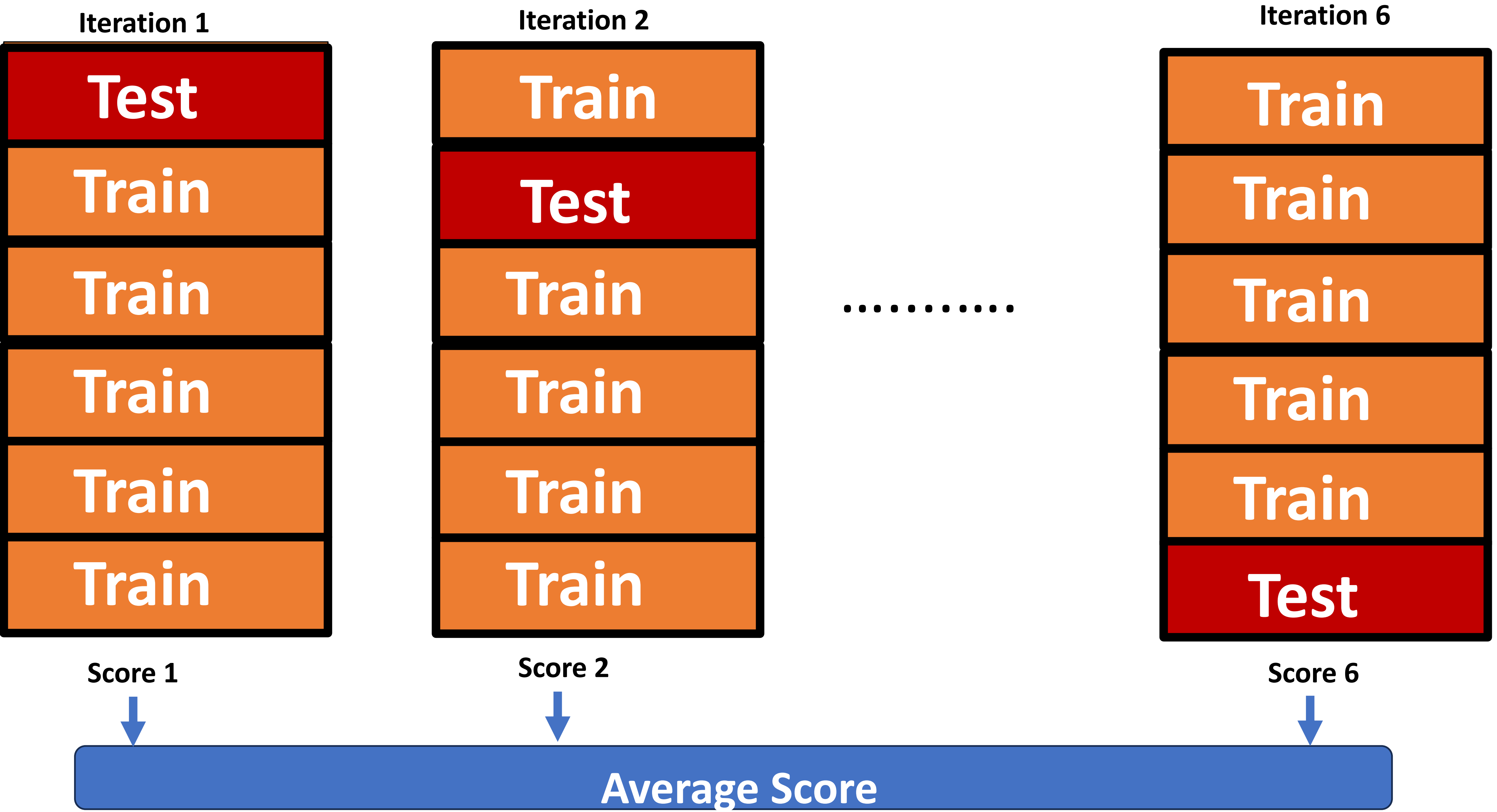
Here, k=6

6-Cross-validation

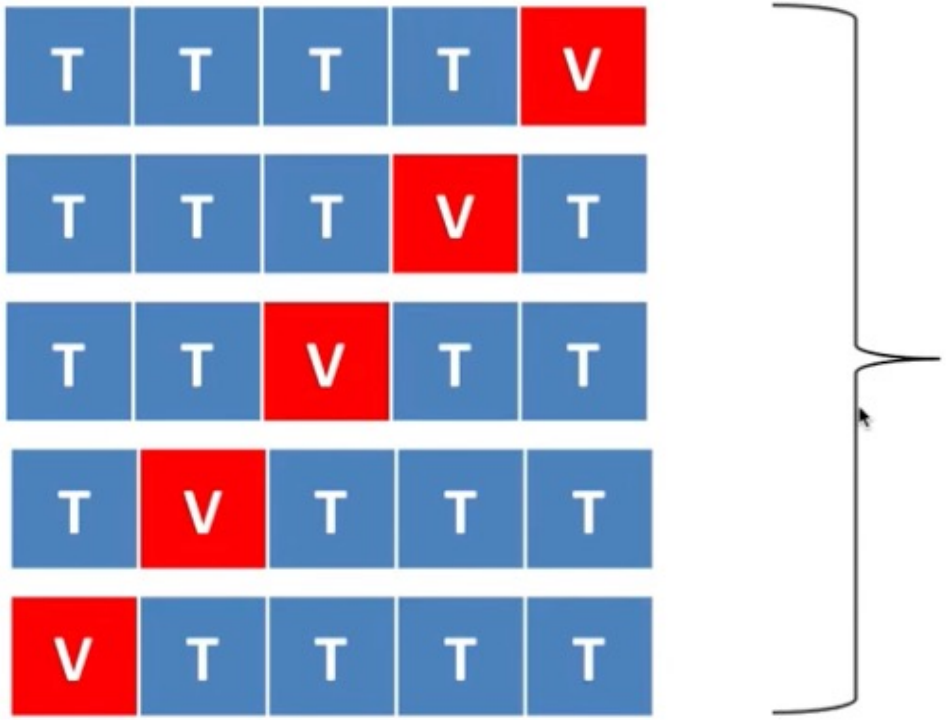
6-fold Cross-validation



6-fold Cross-validation



K-fold Cross-validation



n: number of data points in total

n_k: number of data points in part k

$$\mathrm{CV}_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{MSE}_k$$

https://en.wikipedia.org/wiki/Cross-validation_(statistics)

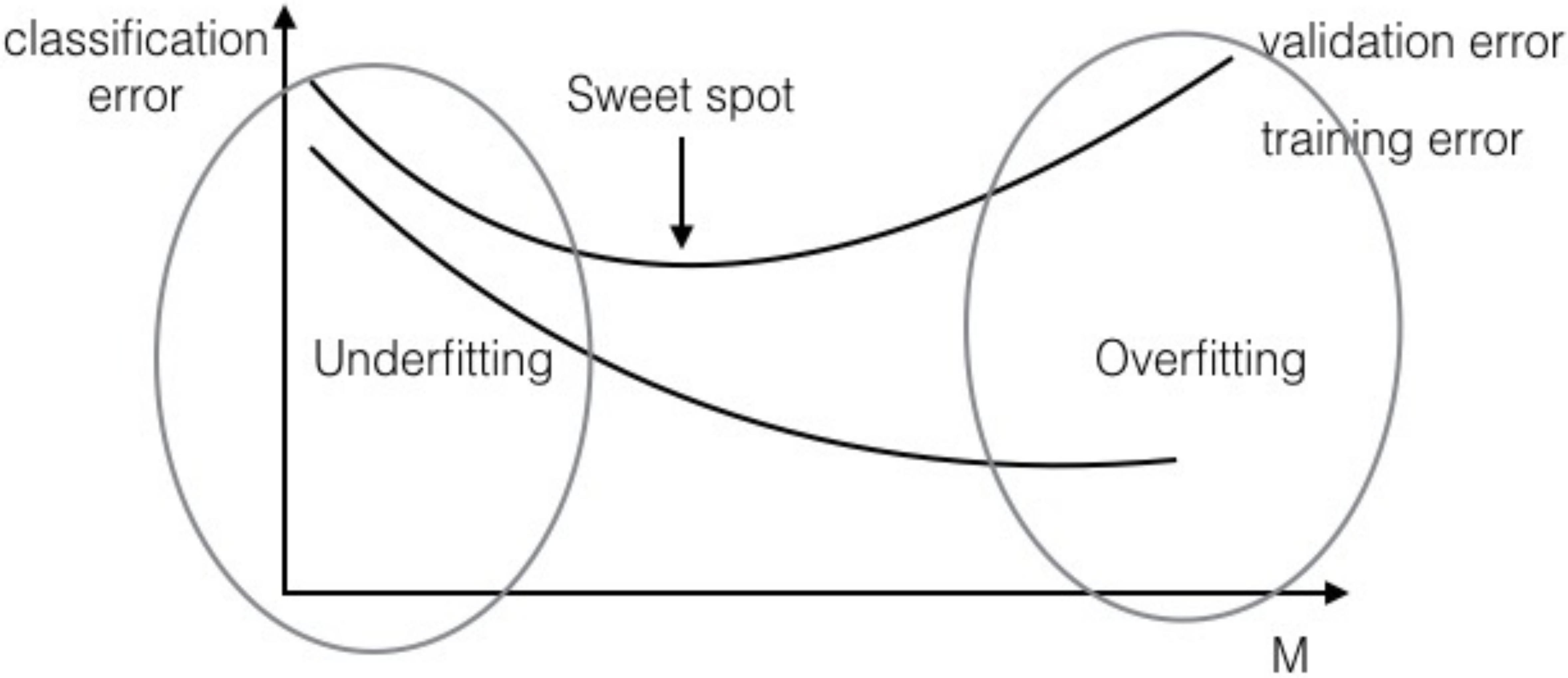
Leave One Out Cross-validation

Setting K = n yields n-fold or Leave-One-Out Cross-Validation (LOOCV).

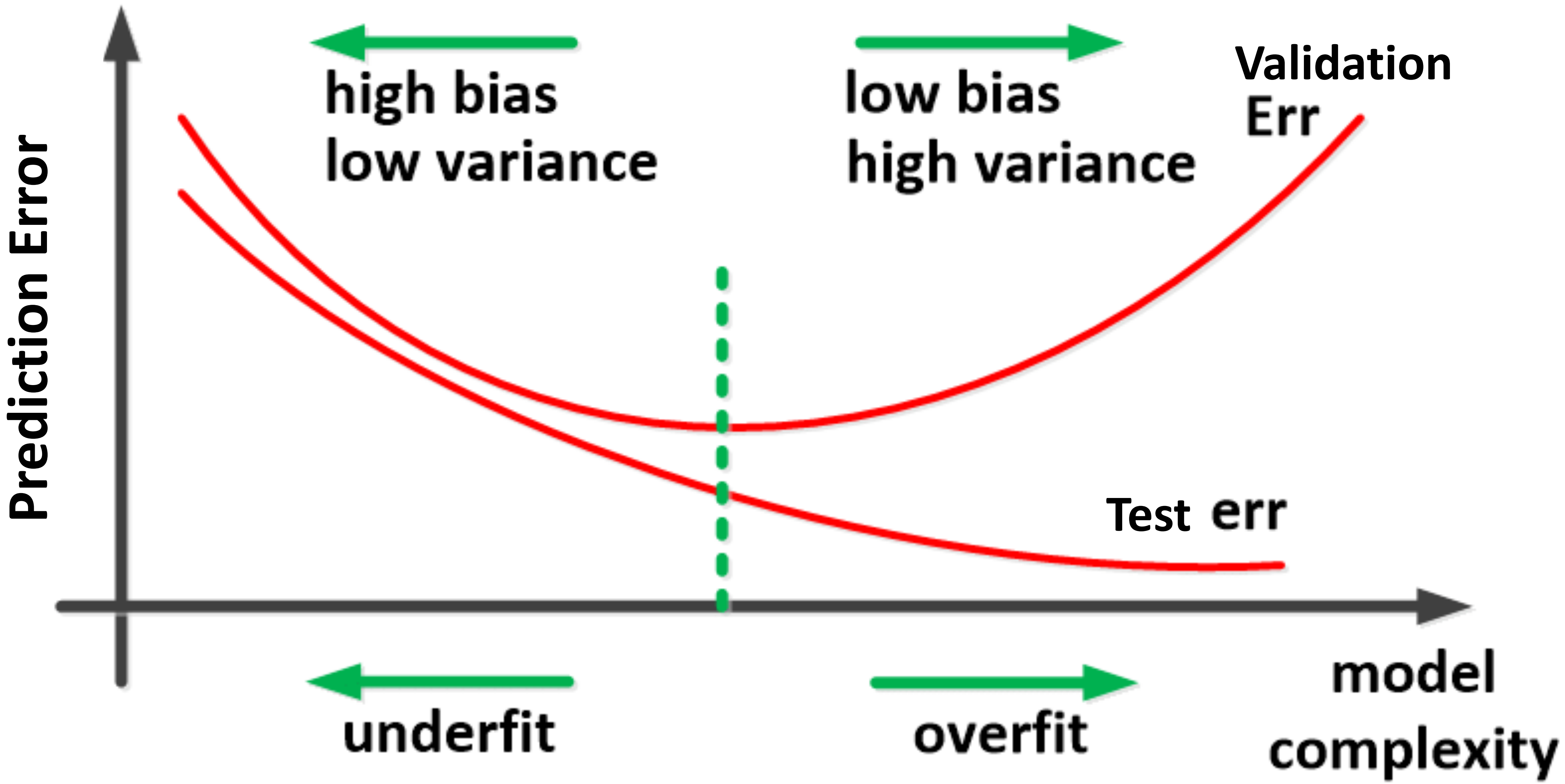
ISSUES WITH CROSS-VALIDATION

- The computational cost increases significantly as we increase the number of folds. LOOCV is much more expensive compared to say, for example, 5-fold or 10-fold.
- This bias is minimized when K = n (LOOCV), but this estimate has high variance (i.e. the estimation is too dependent on the training set).
- Typically, K = 5 or 10 provides a good compromise for this bias-variance trade-off.

Training vs Test Error

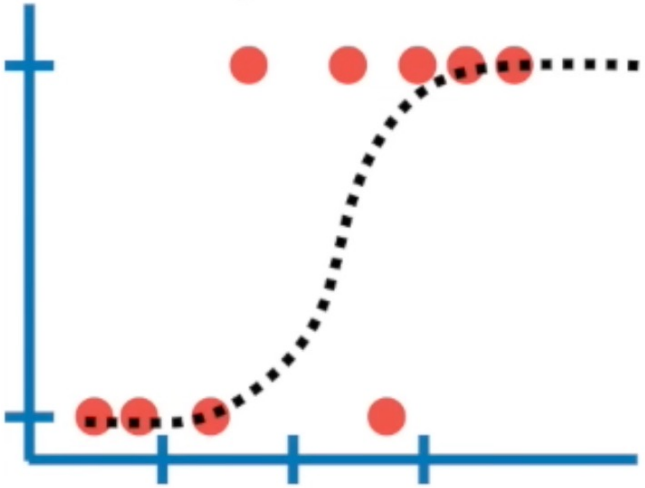


Training vs Test Error

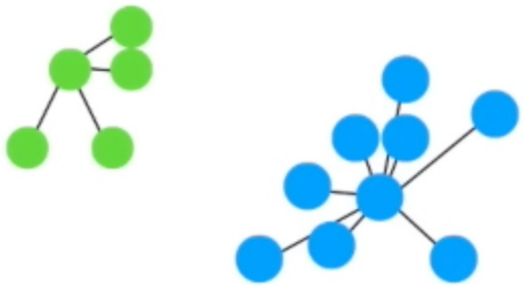


Cross-validation to Compare the ML models

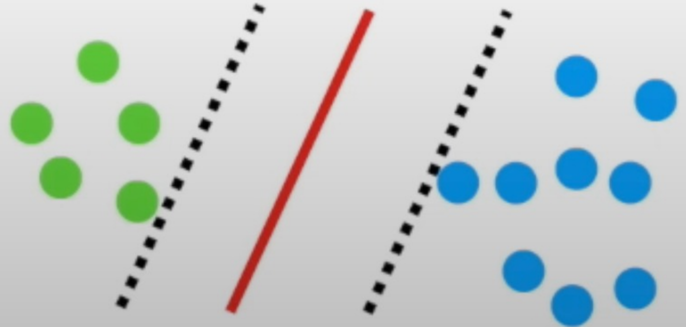
We could use Logistic Regression…



…or K-nearest neighbors…



…or support vector machines (SVM)…



Cross validation allows us to compare different machine learning methods and get a sense of how well they will work in practice.

Cross-validation to Compare the ML models

Logistic Regression	Support Vector Machine	**K-means**
Training Error = 10	Training Error = 12	Training Error = 30
Validation Error = 70	Validation Error = 50	Validation Error = 32

<u>**Which model would you choose?**</u>

Python Implementation

scikit-learn has its own K-fold cross-validation implemented in the `KFold` function. See documentation

```
from sklearn.model_selection import KFold
```

```
kf = KFold(n_splits=5)

errors = []

for idx, (train, test) in enumerate(kf.split(X)):
    X_cv_train = X.values[train]
    X_cv_test = X.values[test]

    y_cv_train = y.values[train]
    y_cv_test = y.values[test]


    # Model fit and prediction
    model = lr.fit(X_cv_train,y_cv_train)
    y_pred_test = model.predict(X_cv_test)
    y_pred_train = model.predict(X_cv_train)


    # Computing errors
    rmse_test = np.sqrt(mean_squared_error(y_cv_test, y_pred_test))
    rmse_train = np.sqrt(mean_squared_error(y_cv_train, y_pred_train))

    r2_test = r2_score(y_cv_test, y_pred_test)
    r2_train = r2_score(y_cv_train, y_pred_train)
```

Next Lecture

Forward Models and Databases