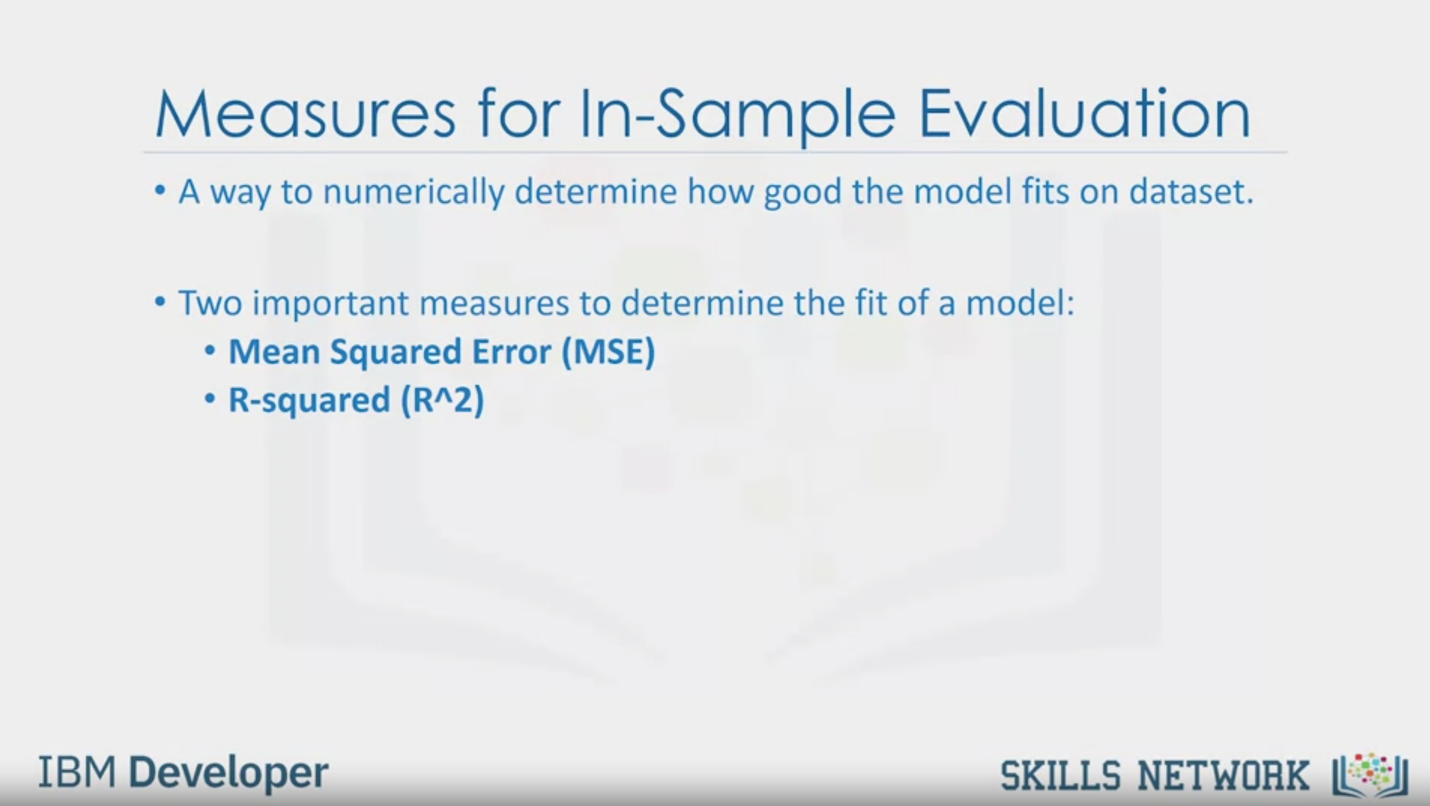


Measures for In-Sample

Evaluation

IBM Developer

SKILLS NETWORK



Measures for In-Sample Evaluation

• A way to numerically determine how good the model fits on dataset.

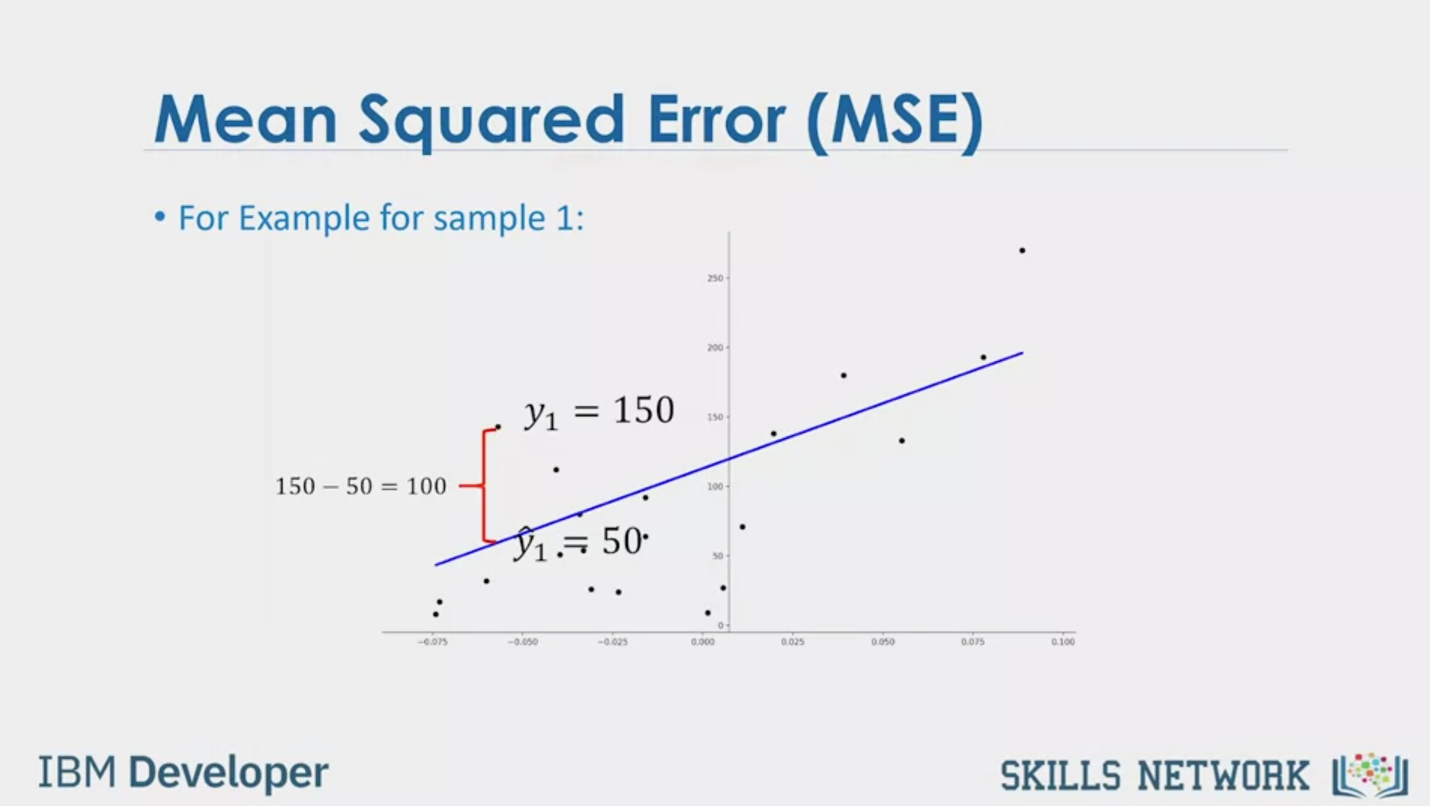
• Two important measures to determine the fit of a model:

• Mean Squared Error (MSE)

• R-squared (R^2)

IBM Developer

SKILLS NETWORK



Mean Squared Error (MSE)

• For Example for sample 1:

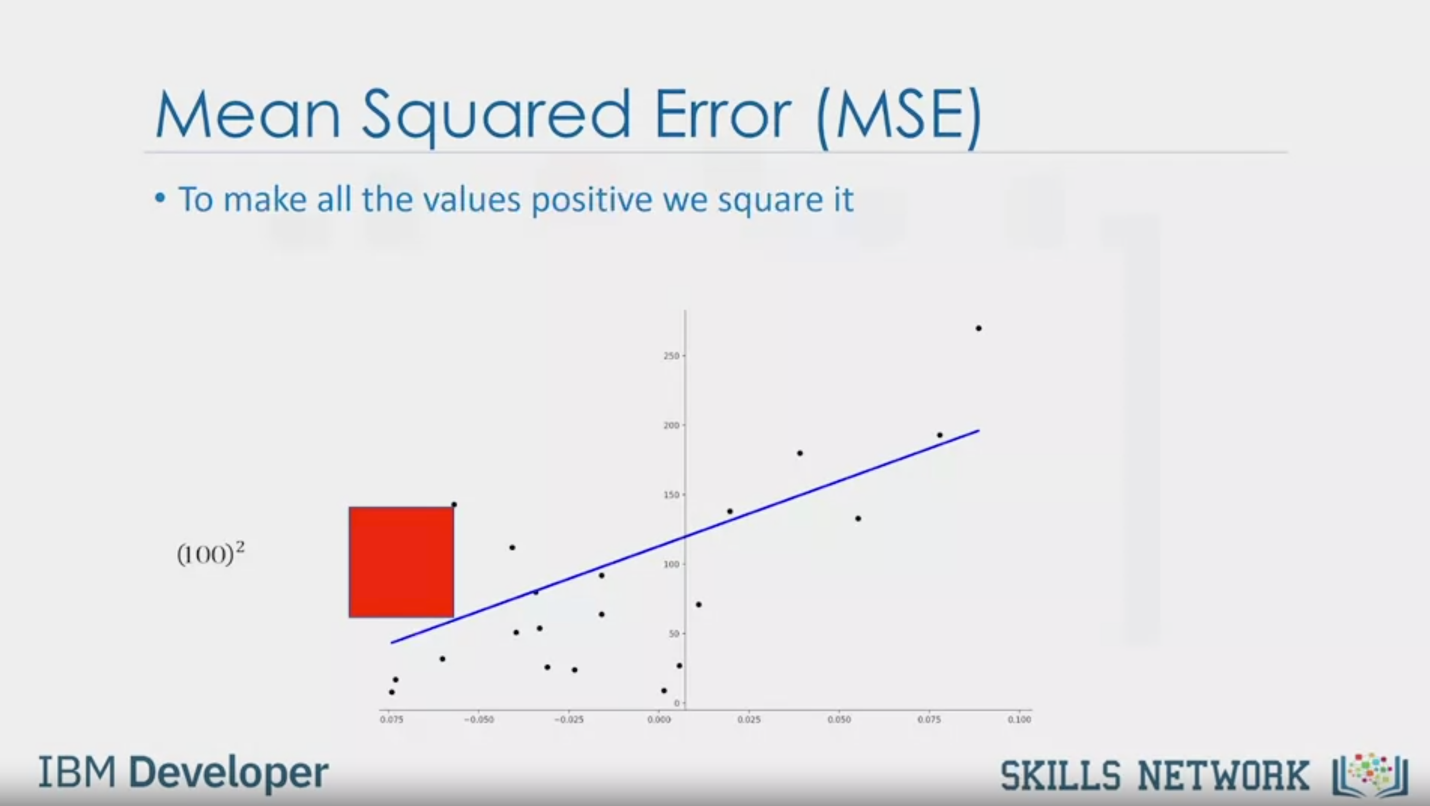
150 – 50 = 100

y1 = 150

y1 = 50

IBM Developer

SKILLS NETWORK



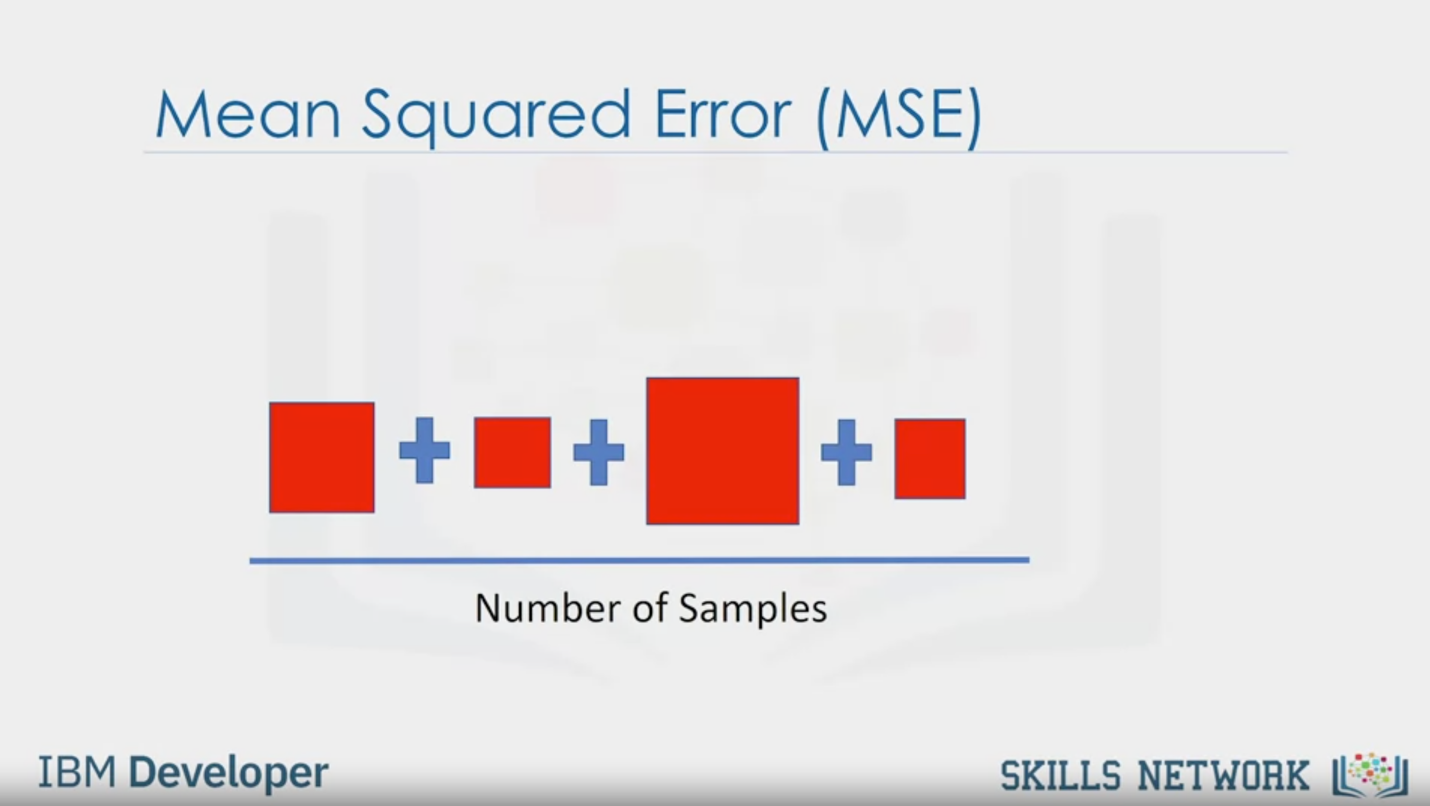
Mean Squared Error (MSE)

• To make all the values positive we square it

(100)2

IBM Developer

SKILLS NETWORK

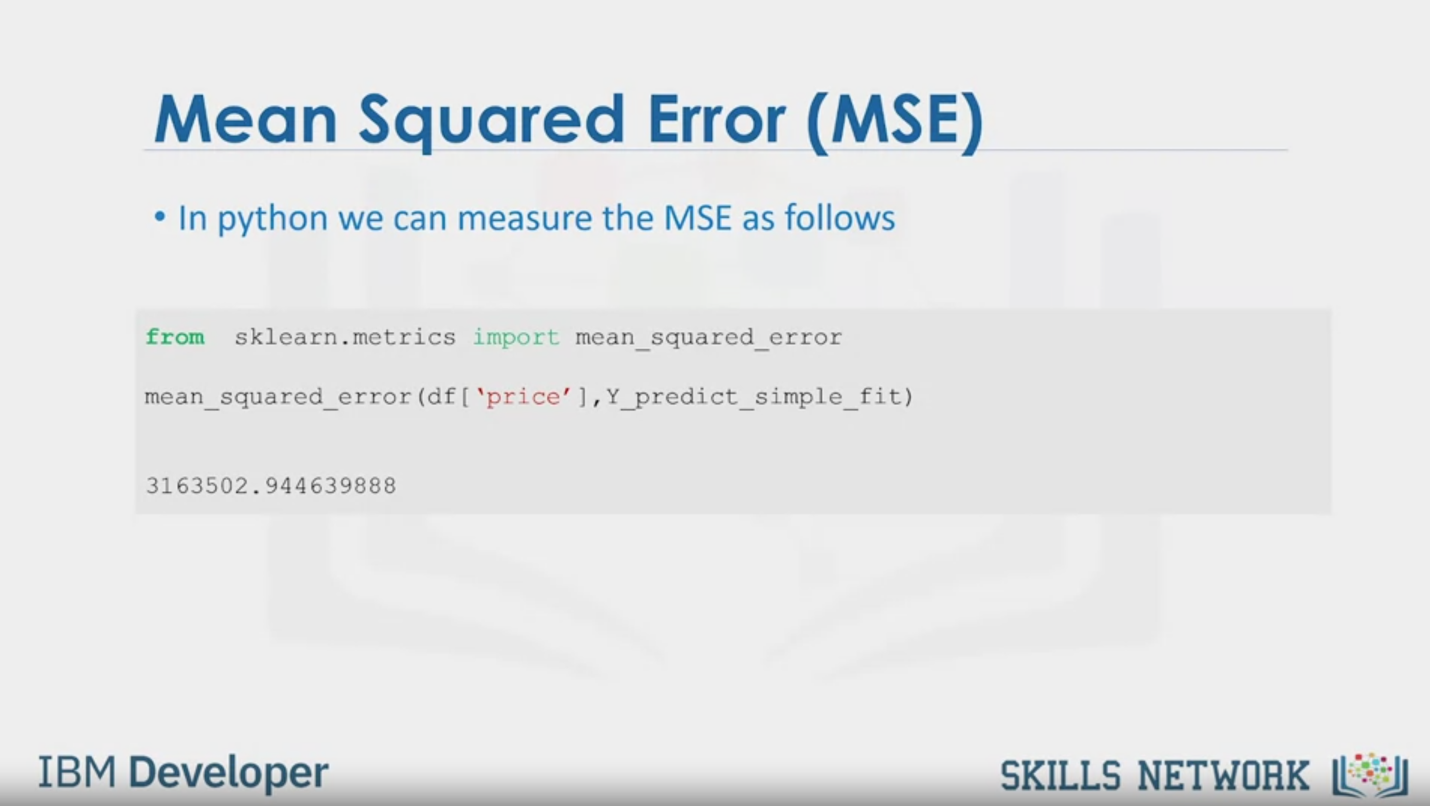


Mean Squared Error (MSE)

Number of Samples

IBM Developer

SKILLS NETWORK



Mean Squared Error (MSE)

• In python we can measure the MSE as follows

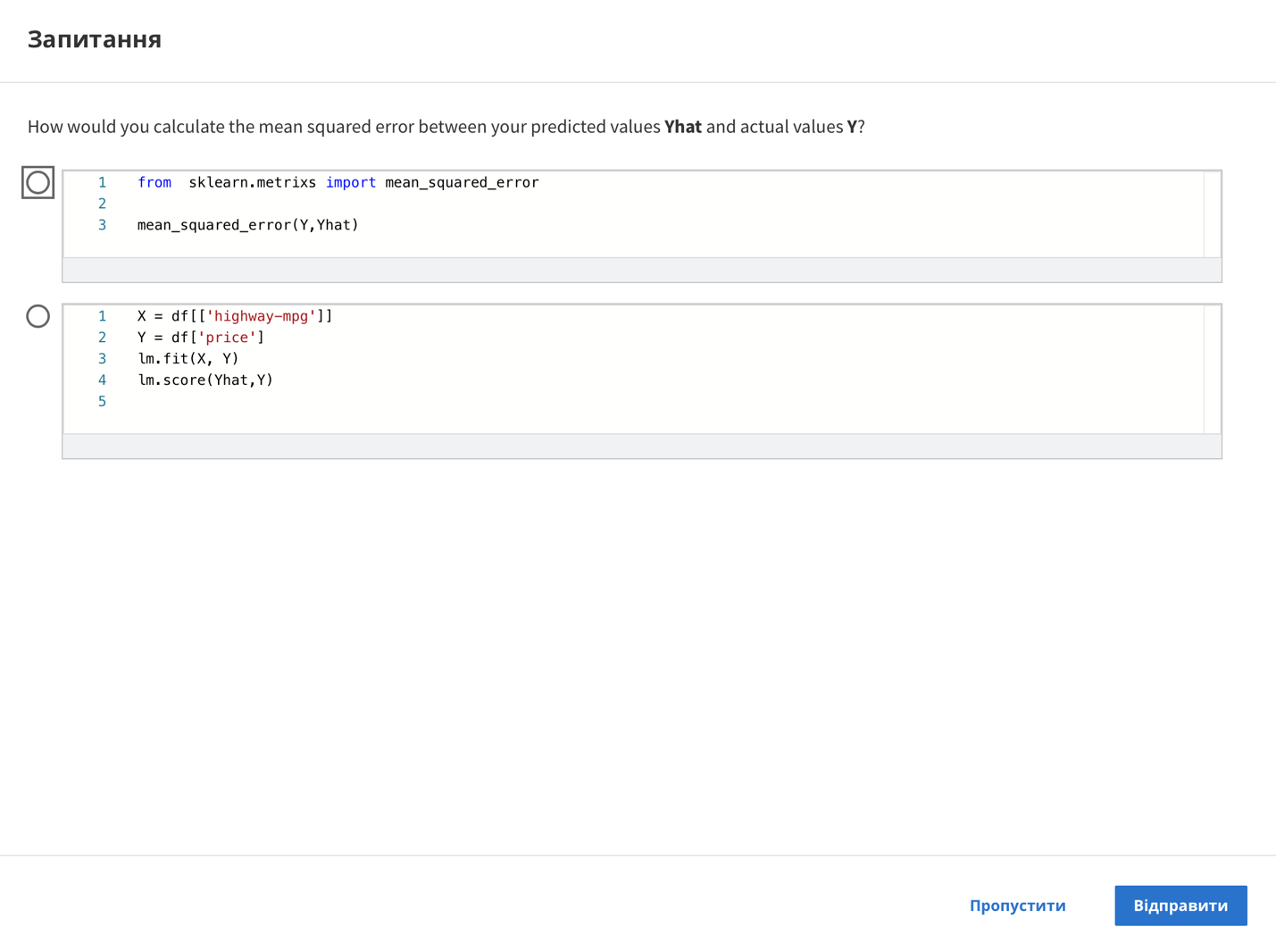
from sklearn.metrics import mean\_squared\_error

mean\_squared\_error (df ['price'], Y\_predict\_simple\_fit)

3163502. 944639888

IBM Developer

SKILLS NETWORK



Question

How would you calculate the mean squared error between your predicted values Yhat and actual values Y?

1 from sklearn.metrixs import\_mean\_squared\_error

2

3 mean\_squared\_error(Y, Yhat)

1 X = df [[ 'highway-mpg' ]]

2 Y = df['price']

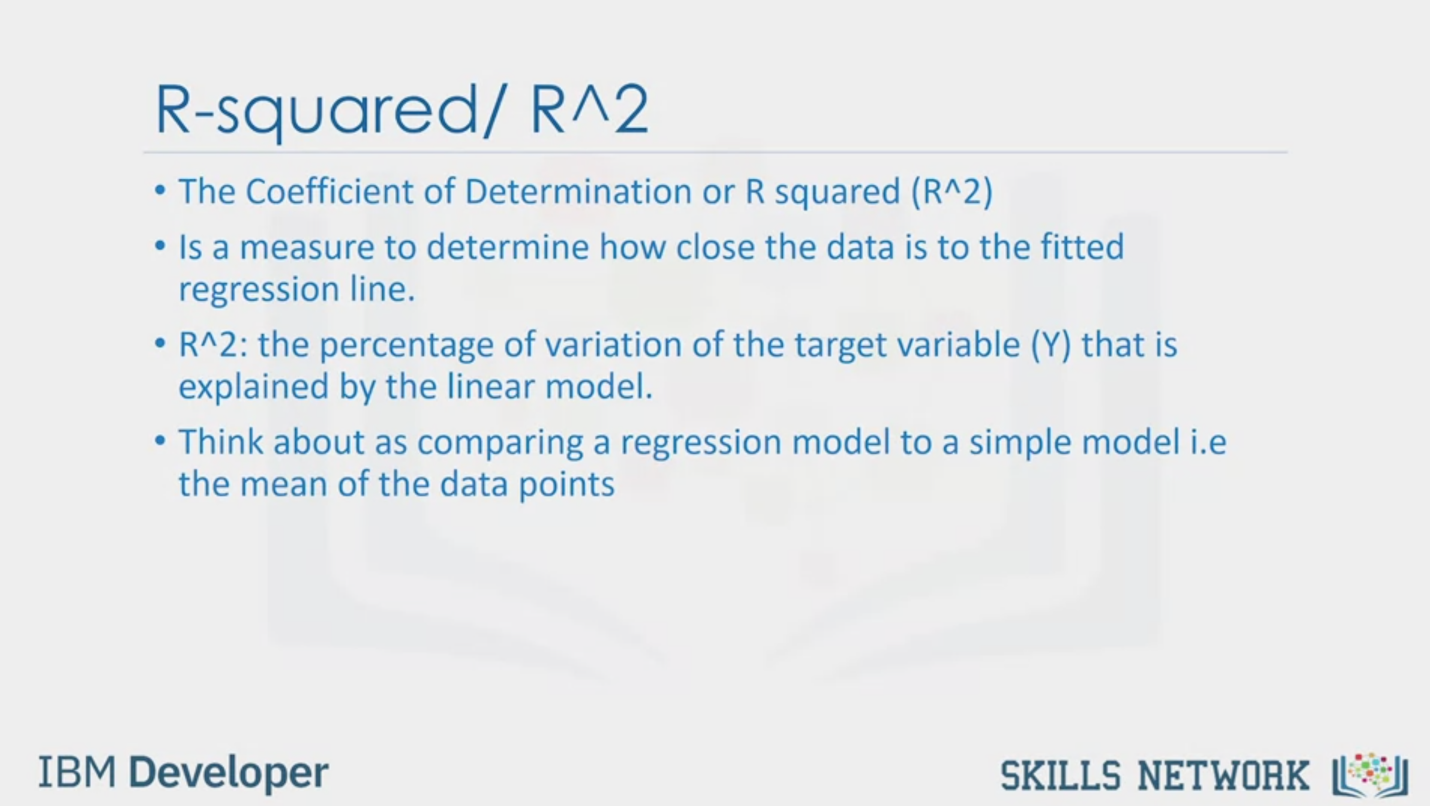
3 lm.fit(X, Y)

4 lm.score(Yhat,Y)

5

Skip

Send



R-squared/R^2

• The Coefficient of Determination or R squared (R^2)

• Is a measure to determine how close the data is to the fitted

regression line.

• R^2: the percentage of variation of the target variable (Y) that is

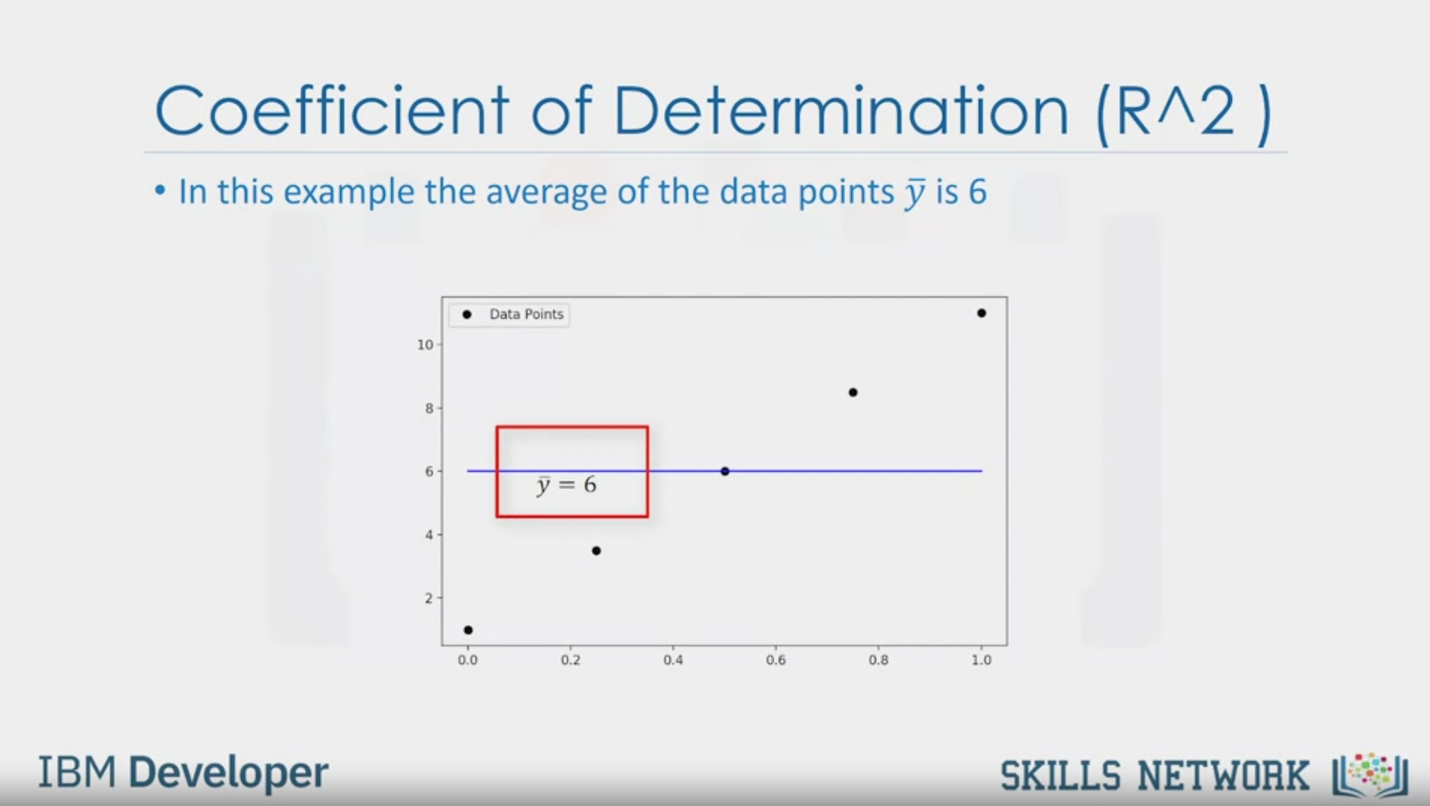
explained by the linear model.

• Think about as comparing a regression model to a simple model i.e

the mean of the data points

IBM Developer

SKILLS NETWORK

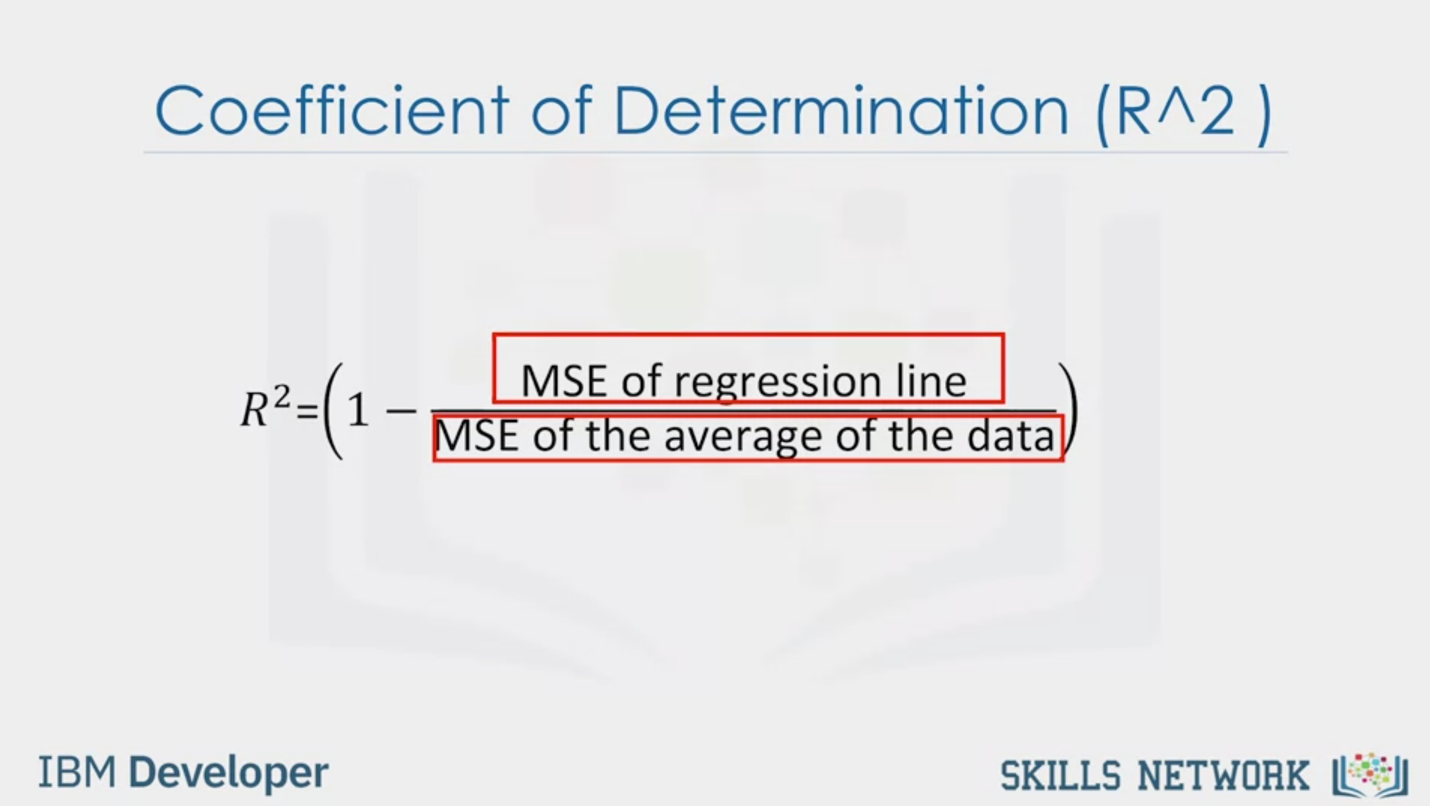


Coefficient of Determination (R^2 )

• In this example the average of the data points is 6

IBM Developer

SKILLS NETWORK

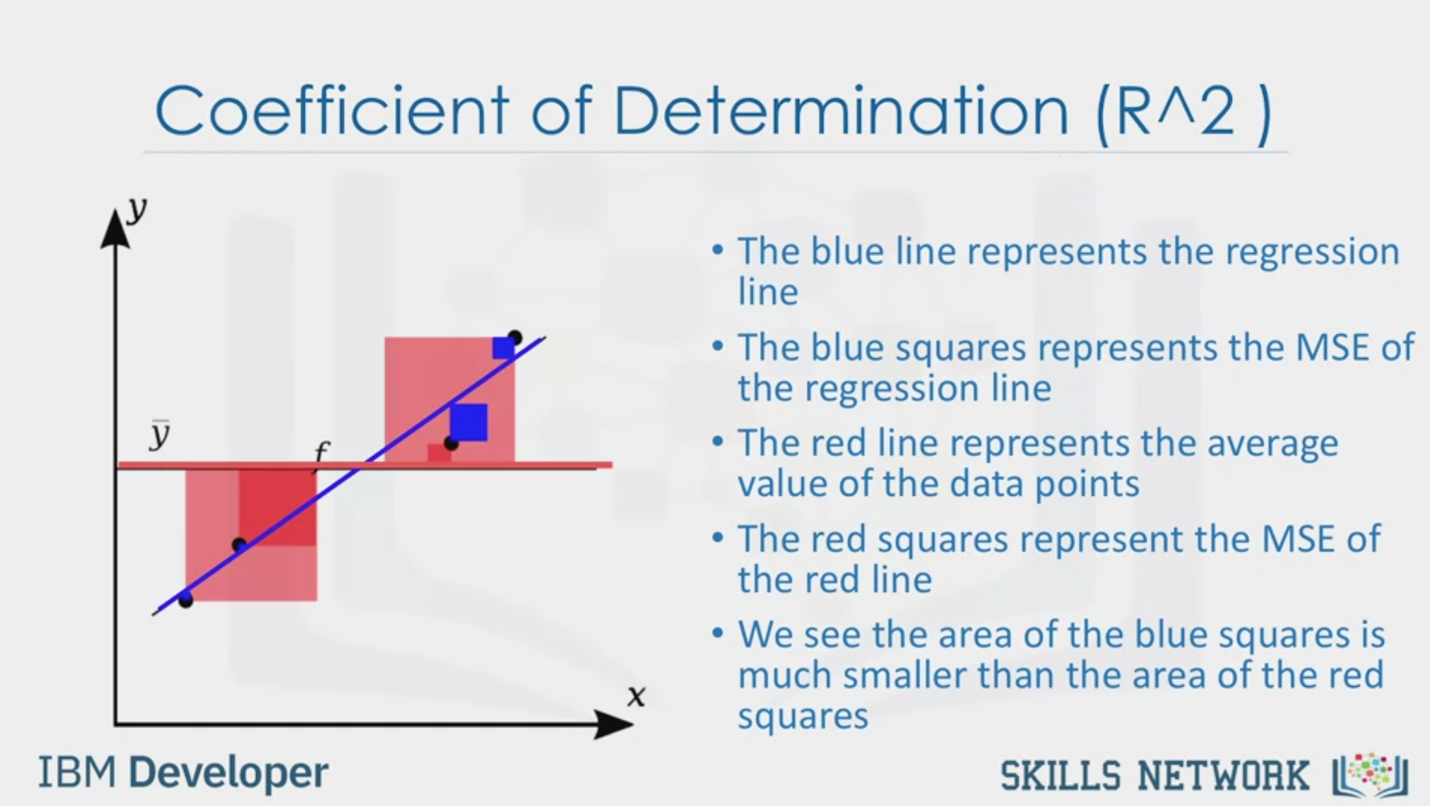


Coefficient of Determination (R^2 )

R2 = (1 - )

IBM Developer

SKILLS NETWORK



Coefficient of Determination (R^2 )

• The blue line represents the regression

line

• The blue squares represents the MSE of

the regression line

• The red line represents the average

value of the data points

• The red squares represent the MSE of

the red line

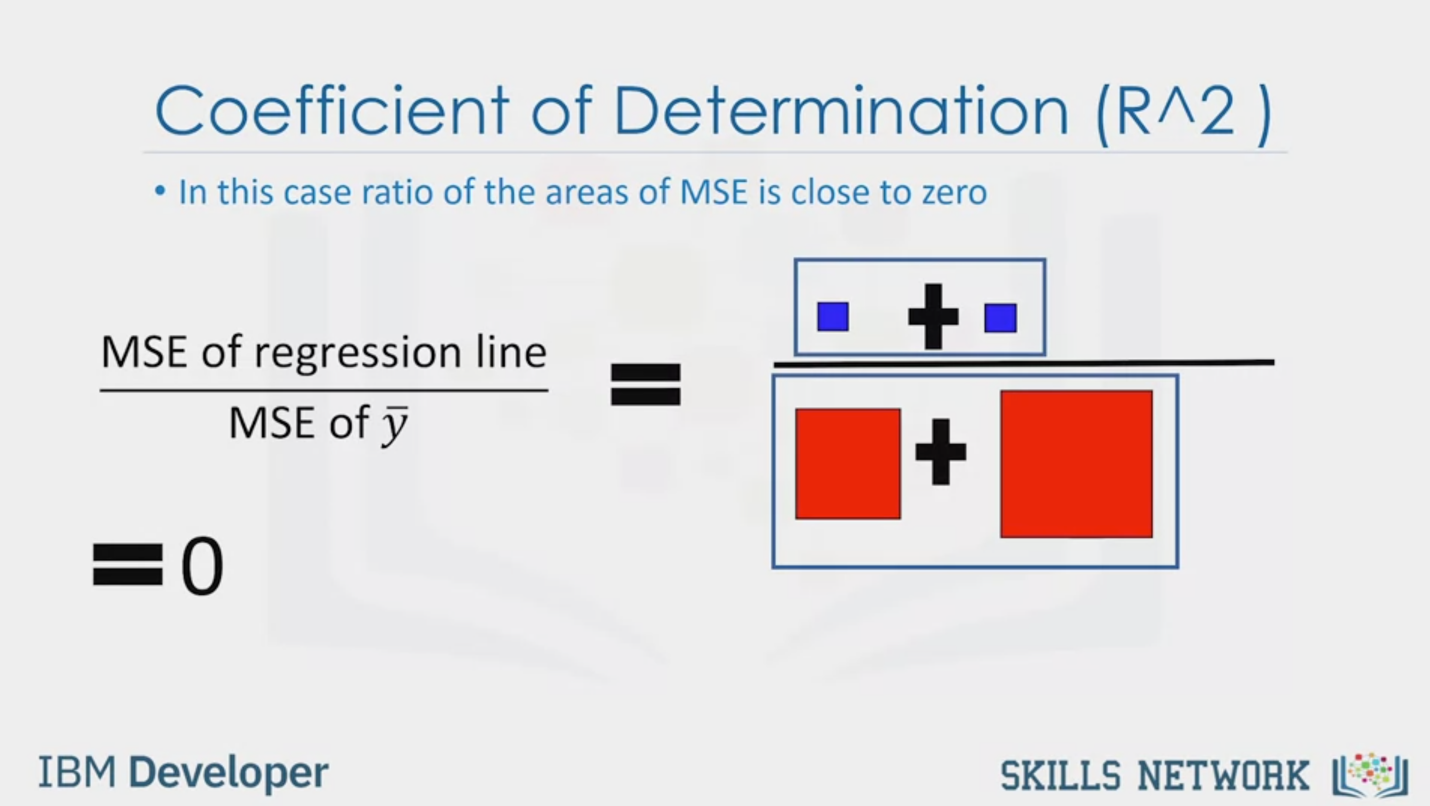
• We see the area of the blue squares is

much smaller than the area of the red

squares

IBM Developer

SKILLS NETWORK

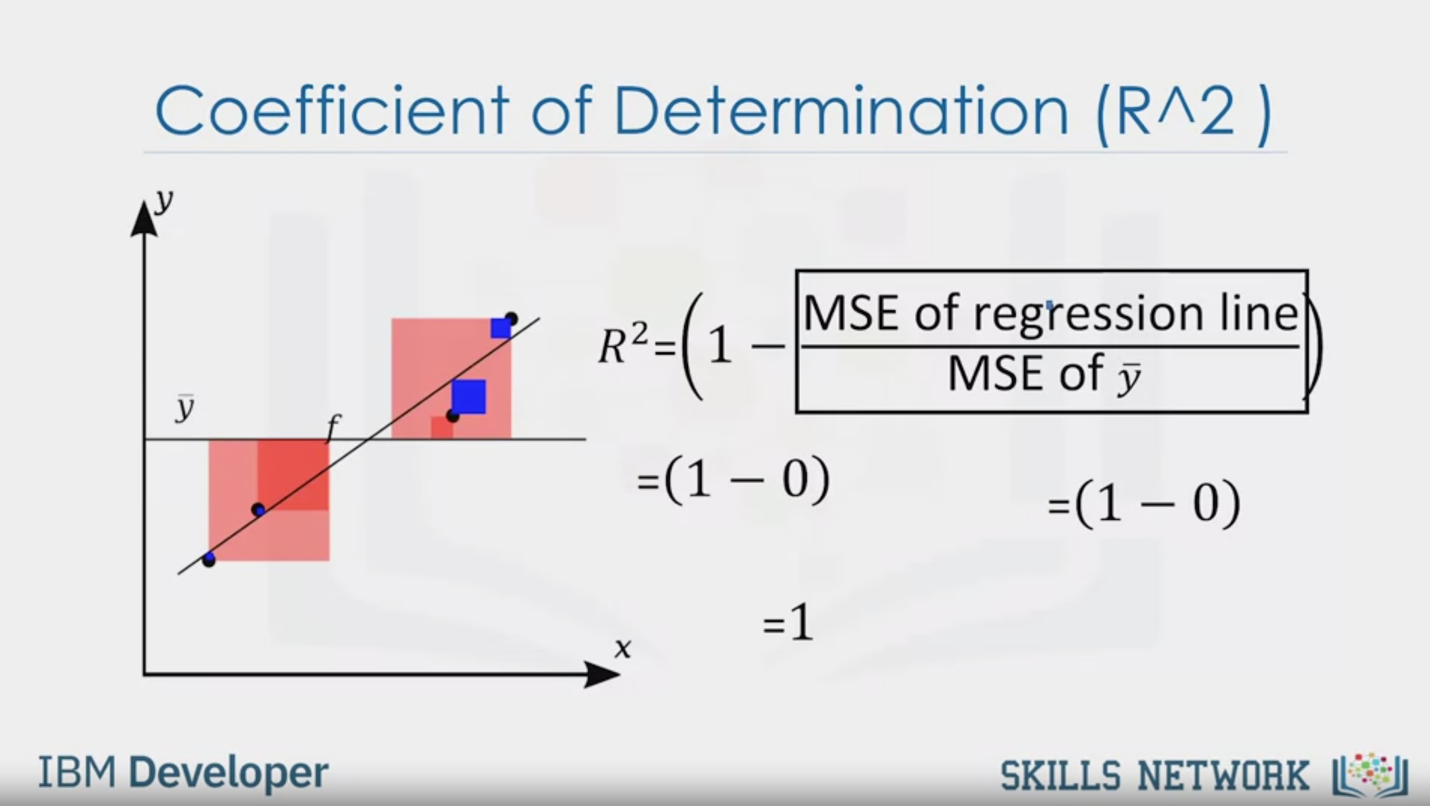


Coefficient of Determination (R^2 )

• In this case ratio of the areas of MSE is close to zero

IBM Developer

SKILLS NETWORK



Coefficient of Determination (R^2 )

R2 = (1 - )

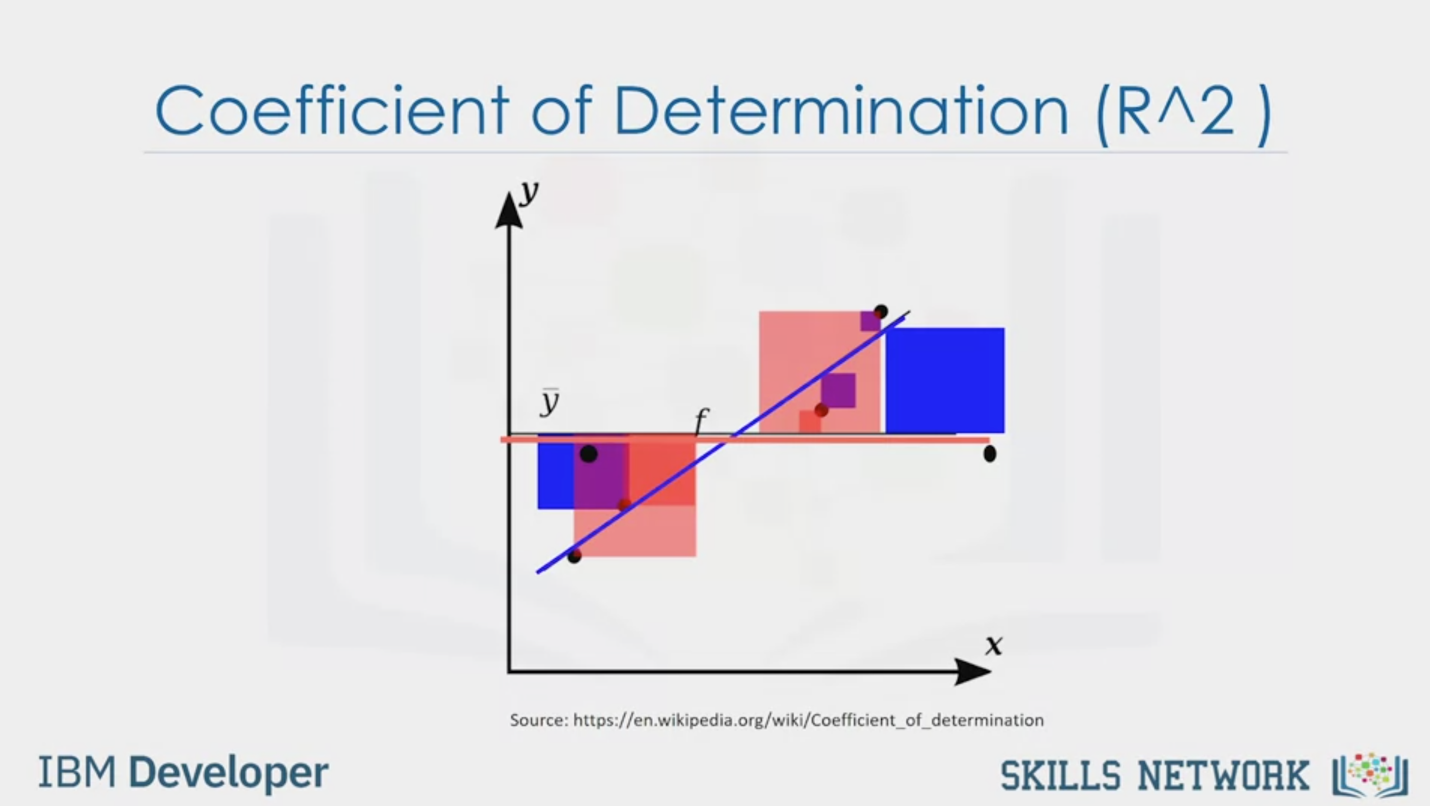
= (1 - 0)

= (1 – 0)

= 1

IBM Developer

SKILLS NETWORK

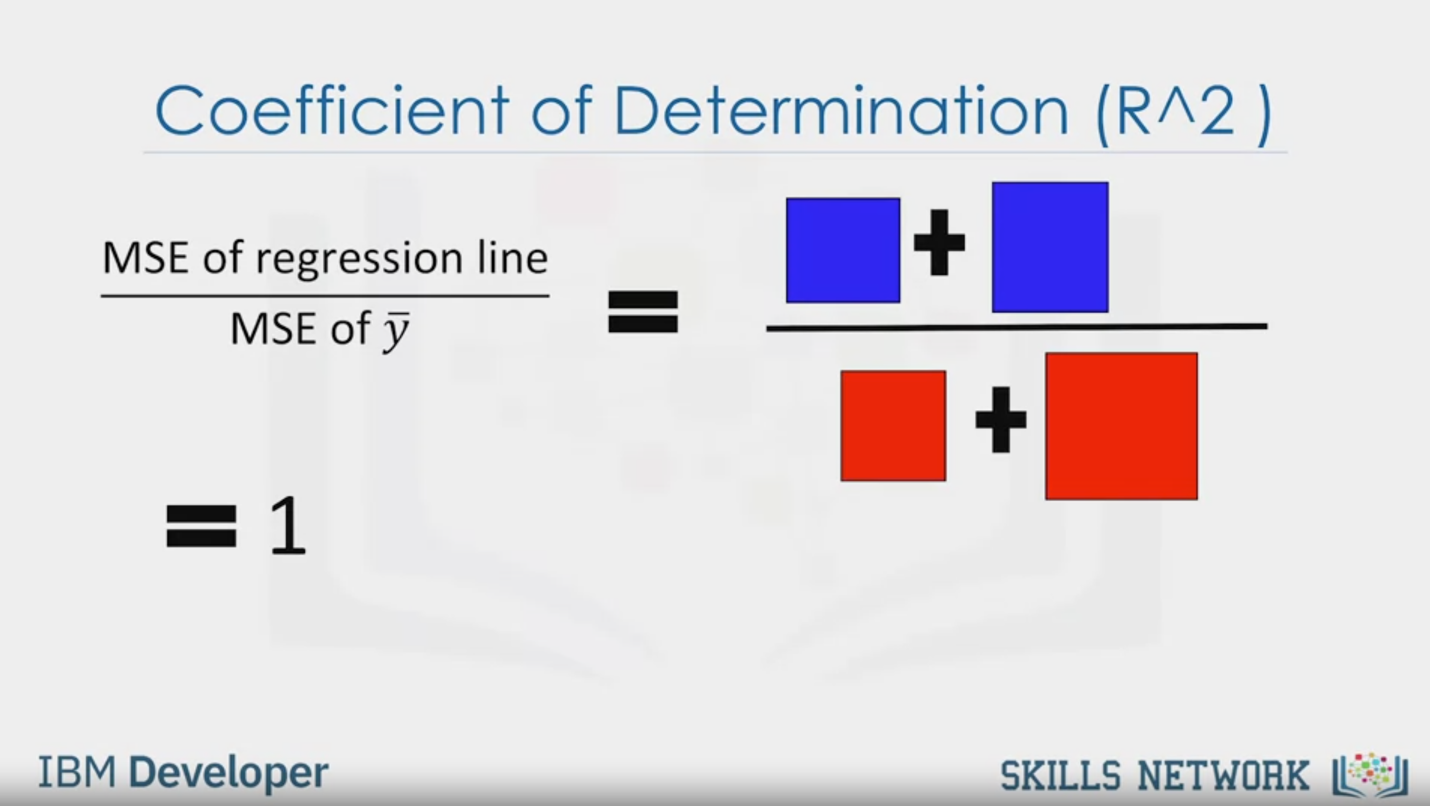


Coefficient of Determination (R^2 )

Source: https://en.wikipedia.org/wiki/Coefficient of determination

IBM Developer

SKILLS NETWORK

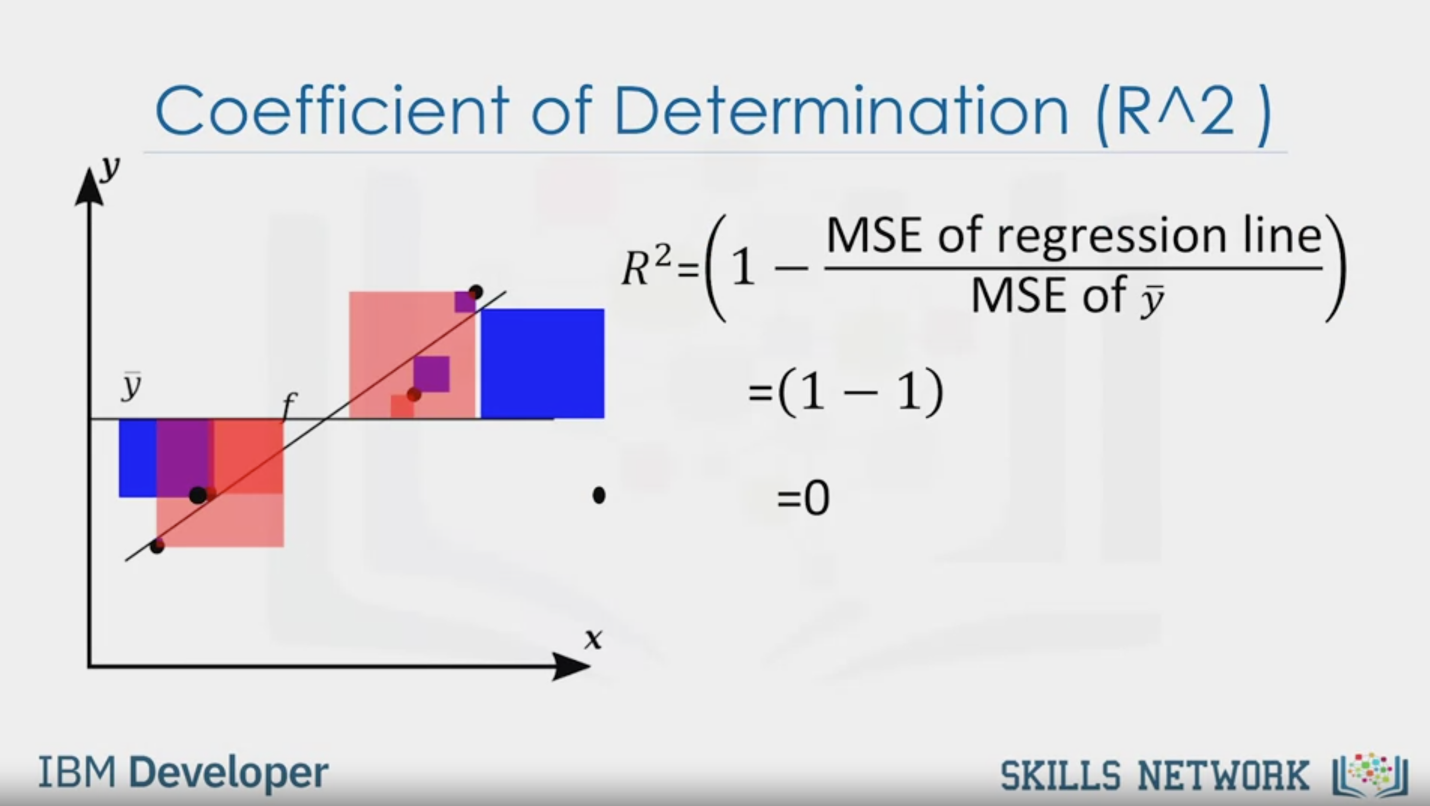


Coefficient of Determination (R^2 )

=1

IBM Developer

SKILLS NETWORK



Coefficient of Determination (R^2 )

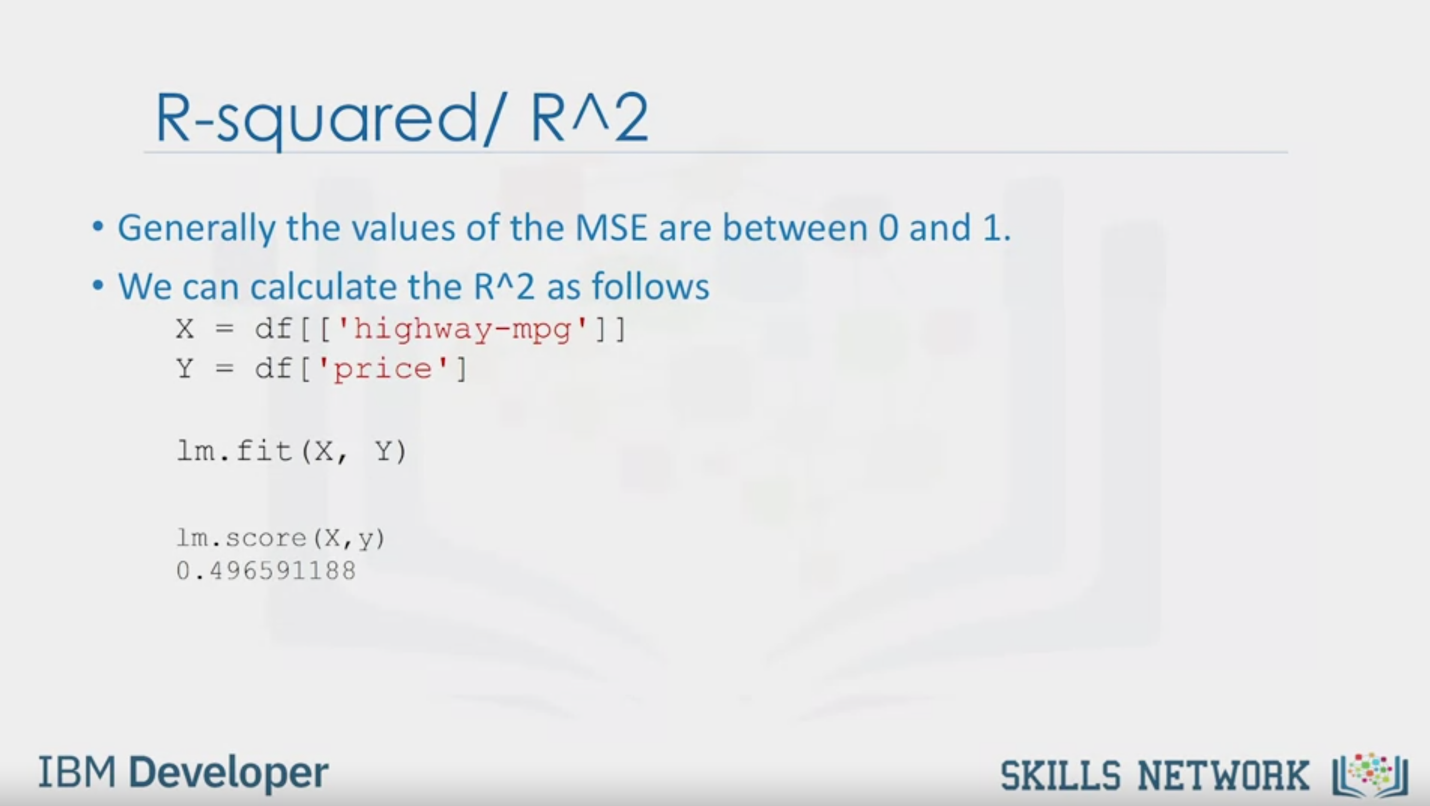
R2 = (1 - )

= (1 - 1)

= 0

IBM Developer

SKILLS NETWORK



R-squared/R^2

• Generally the values of the MSE are between 0 and 1.

• We can calculate the R^2 as follows

X = df[['highway-mpg']]

Y = df['price']

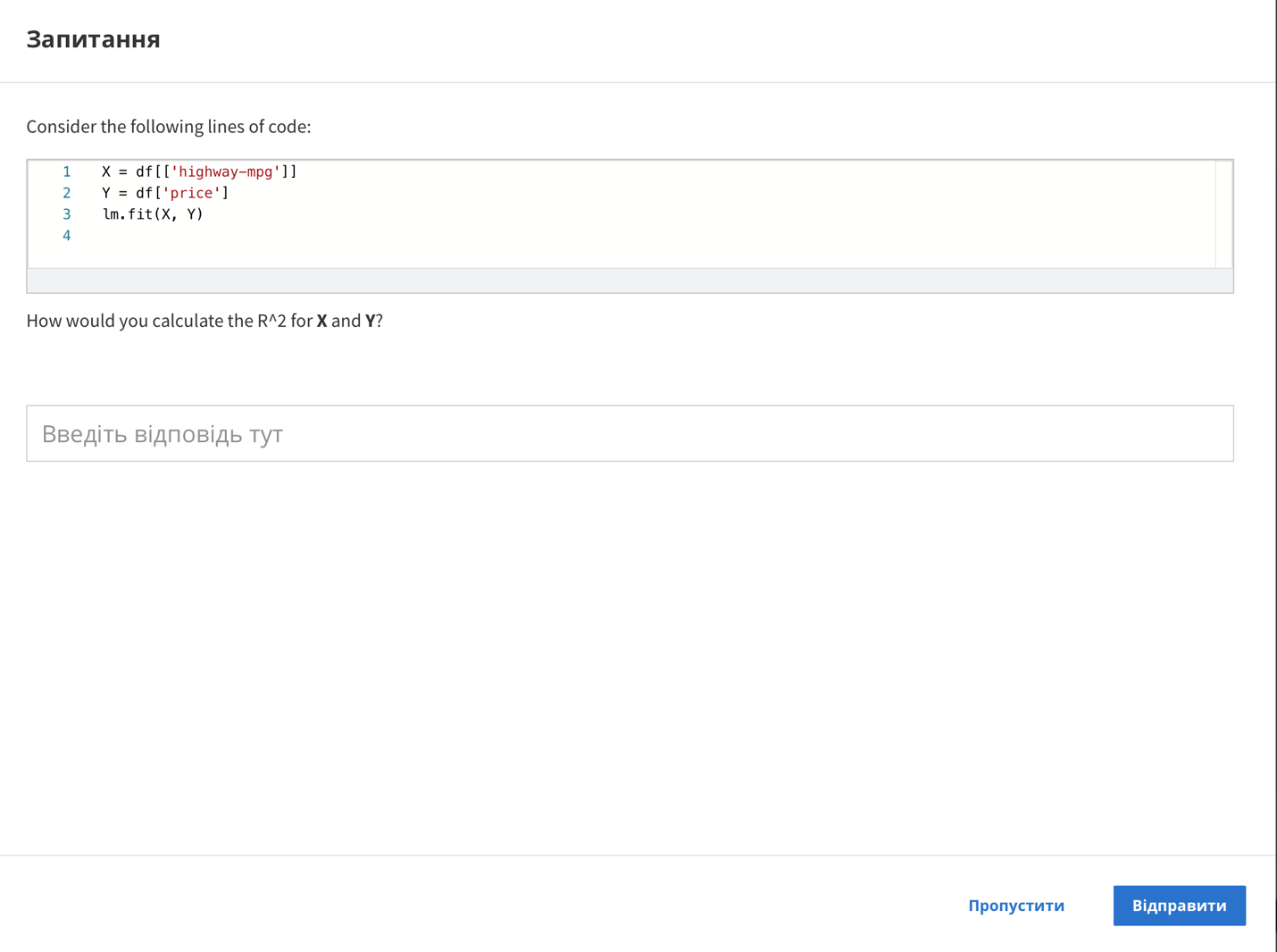
lm.fit(X, Y)

lm.score (X, y)

0.496591188

IBM Developer

SKILLS NETWORK



Question

Consider the following lines of code:

1 X = df[[ ' highway-mpg' ]]

2 Y = df['price']

3 lm.fit(X, Y)

4

How would you calculate the R^2 for X and Y?

Skip

Send



IM Developer

SKILLS NETWORK

Now that we’ve seen how we can evaluate a model by using visualization, we want to

numerically evaluate our models.

Let’s look at some of the measures that we use for in-sample evaluation.

These measures are a way to numerically determine how good the model fits on our data.

Two important measures that we often use to determine the fit of a model are: Mean Square

Error (MSE), and R-squared.

To measure the MSE, we find the difference between the actual value y and the predicted

value yhat then square it.

In this case, the actual value is 150; the predicted value is 50. Subtracting these points

we get 100.

We then square the number.

We then take the Mean or average of all the errors by adding then all together and dividing

by the number of samples.

To find the MSE in Python, we can import the “mean\_Squared\_error” from “scikit-learn.metrics”.

The “mean\_Squared\_error” function gets two inputs: the actual value of target variable

and the predicted value of target variable.

R-squared is also called the coefficient of determination. It’s a measure to determine

how close the data is to the fitted regression line. So how close is our actual data to our

estimated model?

Think about it as comparing a regression model to a simple model, i.e., the mean of the data

points. If the variable x is a good predictor our model should perform much better than

just the mean.

In this example the average of the data points 𝑦 | is 6.

Coefficient of Determination R^2 is 1 minus the ratio of the MSE of the regression

lined divided by the MSE of the average of the data points. For the most part, it takes

values between 0 and 1.

Let’s look at a case where the line provides a relatively good fit.

The blue line represents the regression line.

The blue squares represent the MSE of the regression line.

The red line represents the average value of the data points.

The red squares represent the MSE of the red line.

We see the area of the blue squares is much smaller than the area of the red squares.

In this case, because the line is a good fit, the Mean squared error is small, therefore

the numerator is small.

The Mean squared error of the line is relatively large, as such the numerator is large.

A small number divided by a larger number is an even smaller number. Taken to an extreme

this value tends to zero.

If we Plug in this value from the previous slide for R^2, we get a value near one, this

means the line is a good fit for the data. Here is an example of a line that does not

fit the data well.

If we just examine the area of the red squares compared to the blue squares, we see the area

is almost identical.

The ratio of the areas is close to one.

In this case the R^2 is near zero.

This line performs about the same as just using the average of the data points, therefore,

this line did not perform well.

We find the R-squared value in Python by using the score method, in the linear regression

object.

From the value that we get from this example, we can say that approximately 49.695% of the

variation of price is explained by this simple linear model.

Your R^2 value is usually between 0 and 1. If your R^2 is negative, it can be due to over fitting

that we will discuss in the next module.