Hyperconnect Pre-task

---

1. [Machine Learning] This question is about linear regression. Use as much as mathematical rigor as you wish.

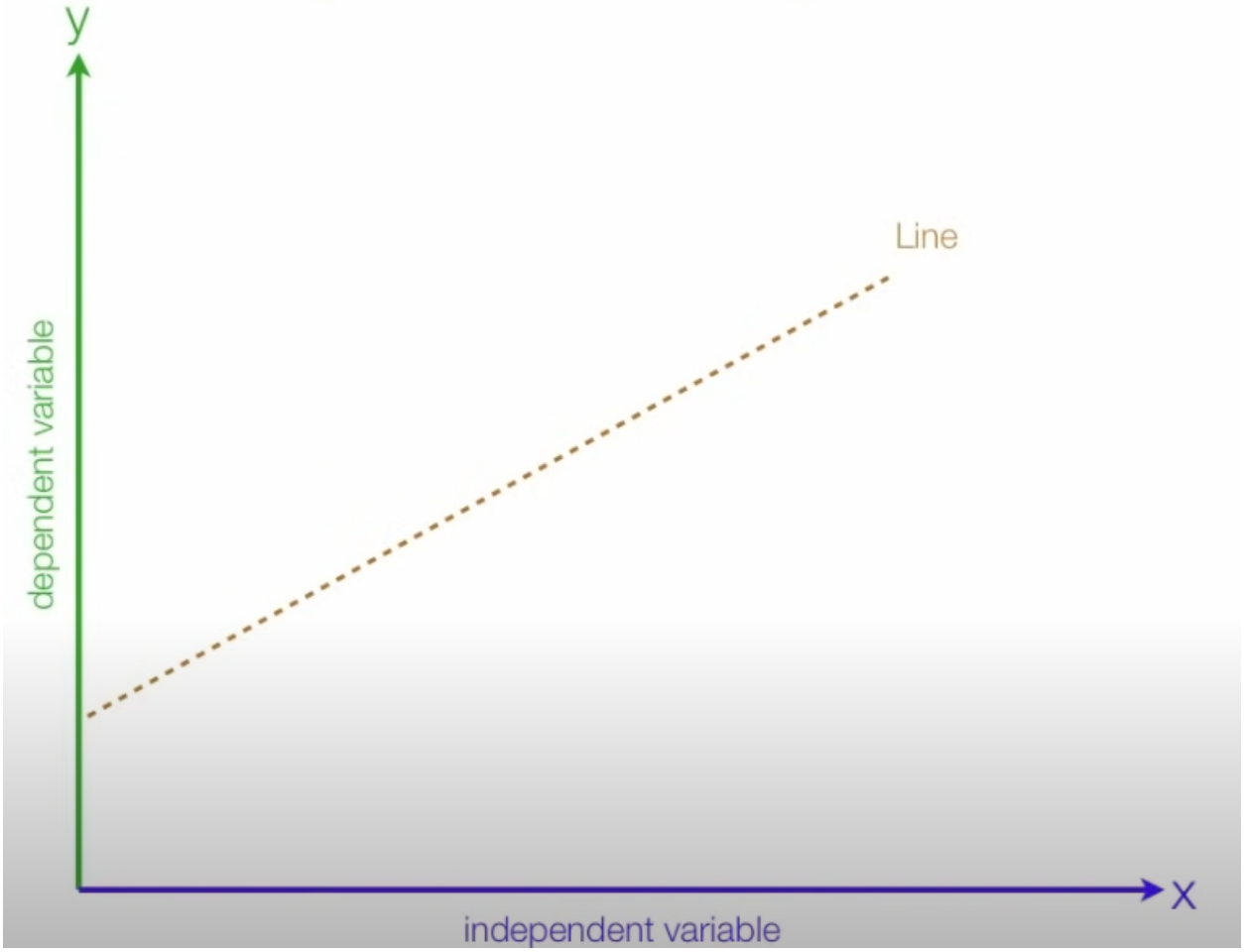
a. What is a linear regression?

Linear Regression

Definition

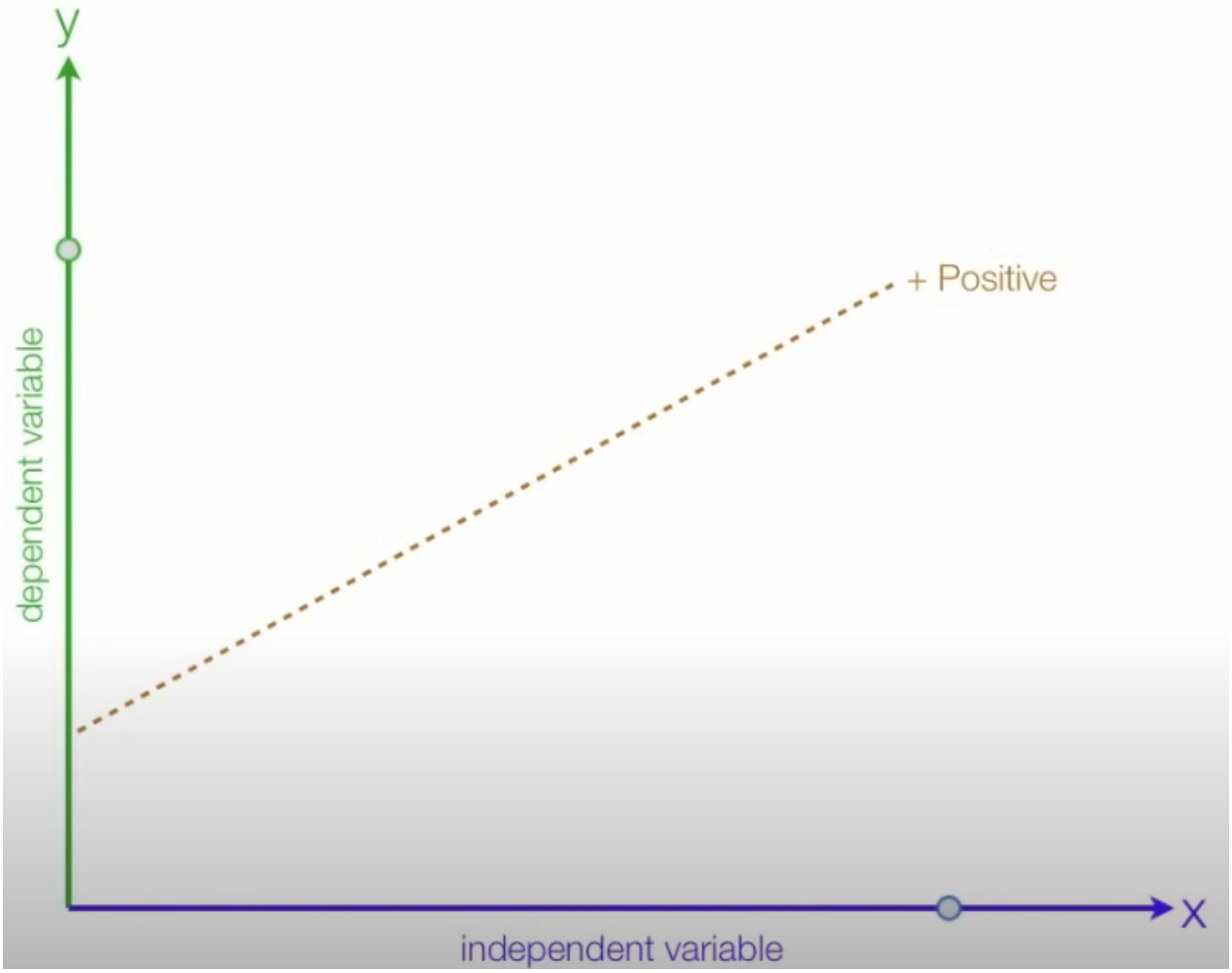
In simple terms, **linear regression** is a technique which is used to find a relationship (a fit) between the `independent variable` and `dependent variable`.

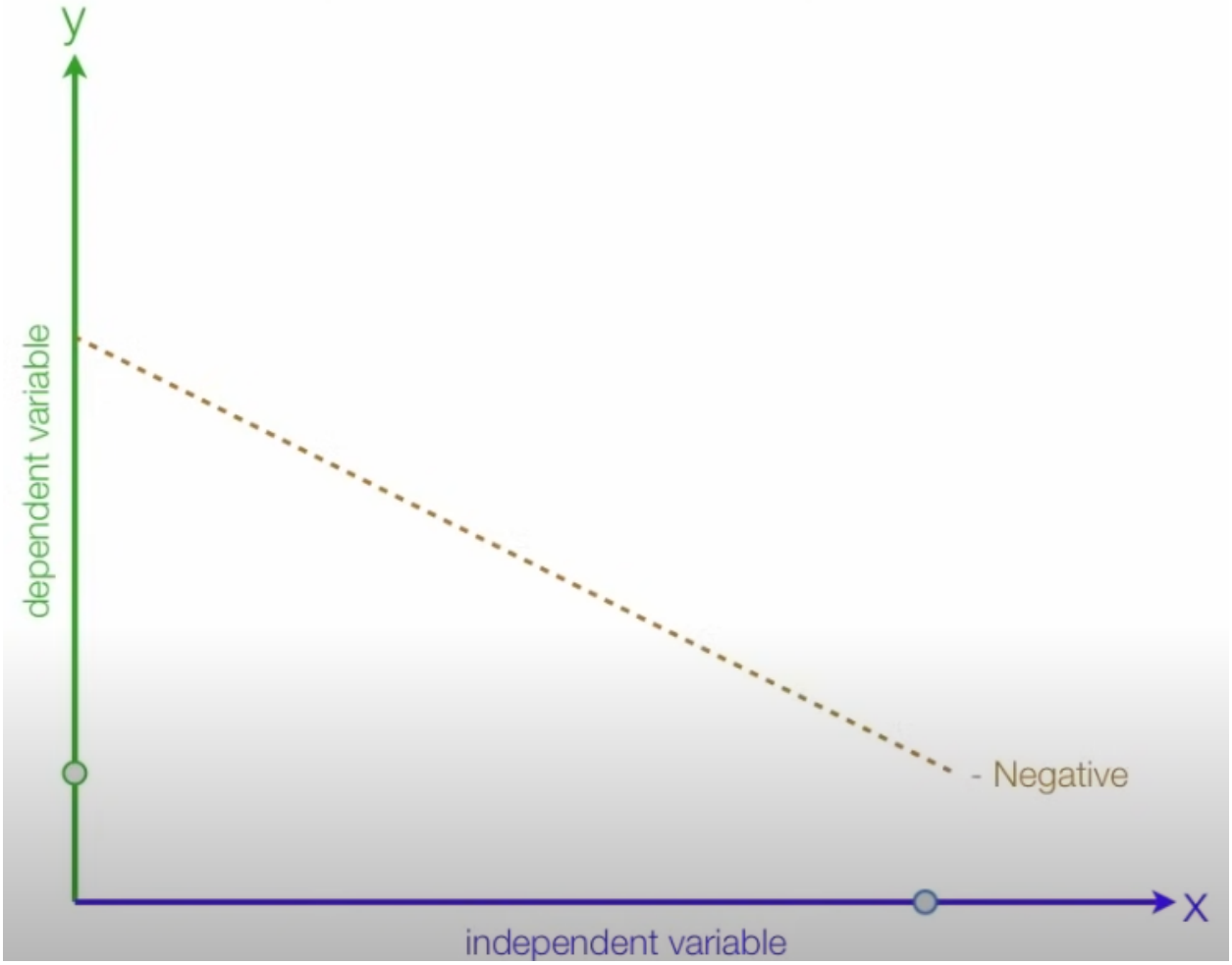
We do this by drawing a *line* between these variables. In this case it is a straight line and hence the name **Linear Regression**.



What we try to understand is that as the `independent variable` is changing or moving, what happens to the `dependent variable`. Does it go up? Does it go down? How does it change?

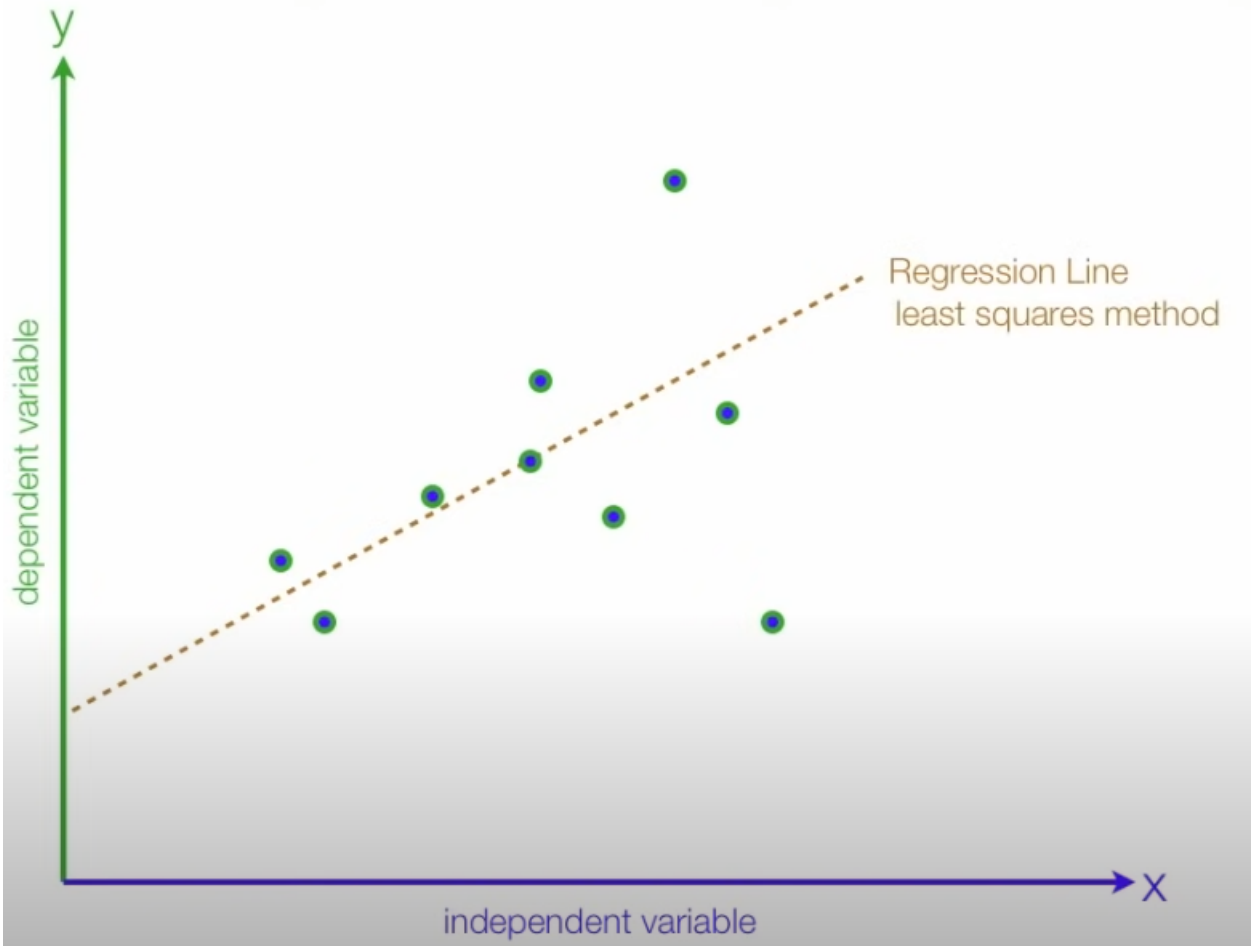
If they move in the same direction i.e. dependent variable increases as the independent variable increase, then they have a **positive relationship.** If the dependent variable decreases as the independent variable increases or vice versa, then the two variables have a **negative relationship.**


y
+ Positive
dependent variable
X
independent variable

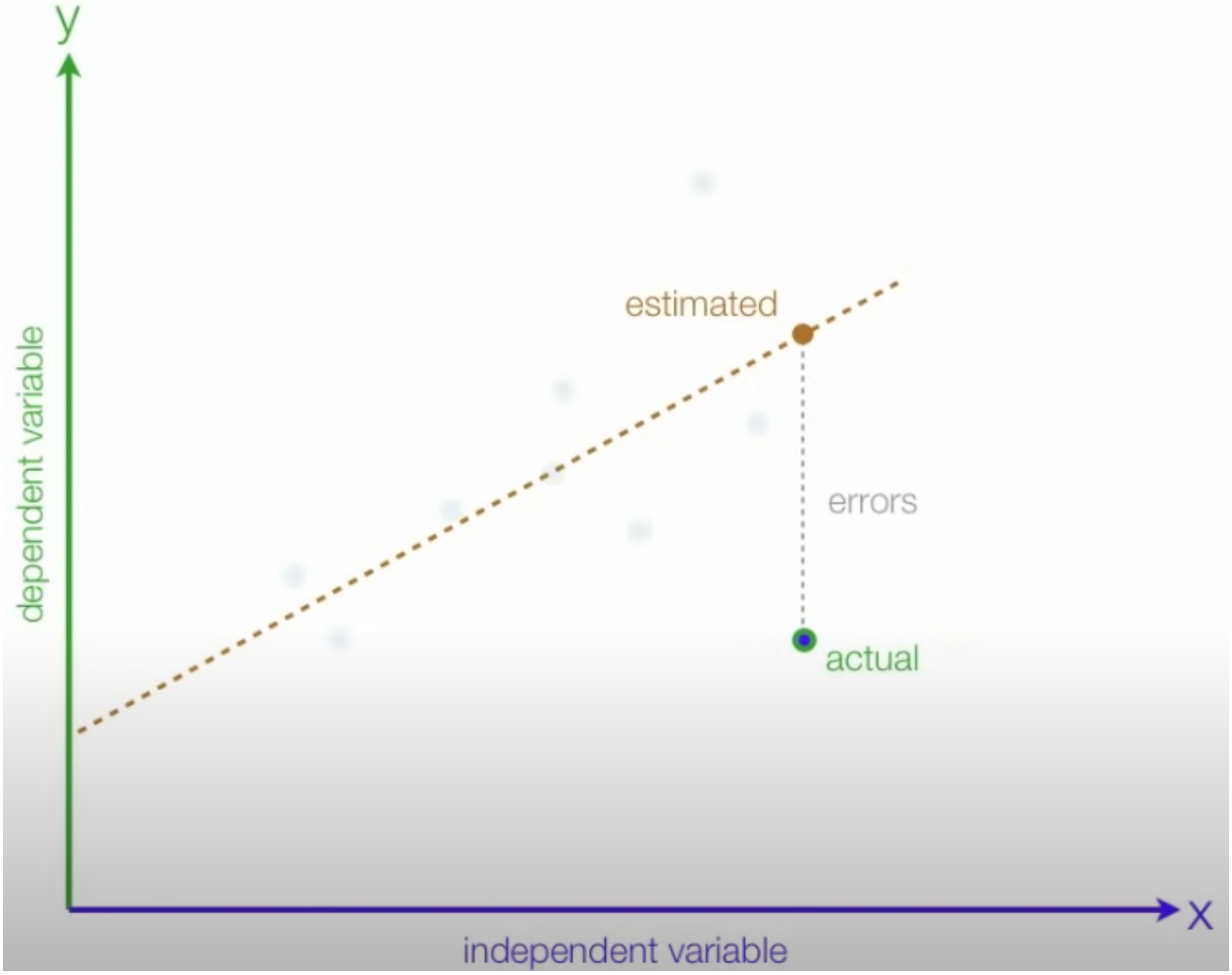


How it works?

To actual conduct the **regression** we collect some `observations` and plot them. Then we try to find a straight **line** that will fits through all these different observation points. This line is called **Regression Line** and it is based upon the `least squares method`.



In the end, we want to **minimize** the distance between the `estimated` values and the `actual` values. This distance is called `error`.



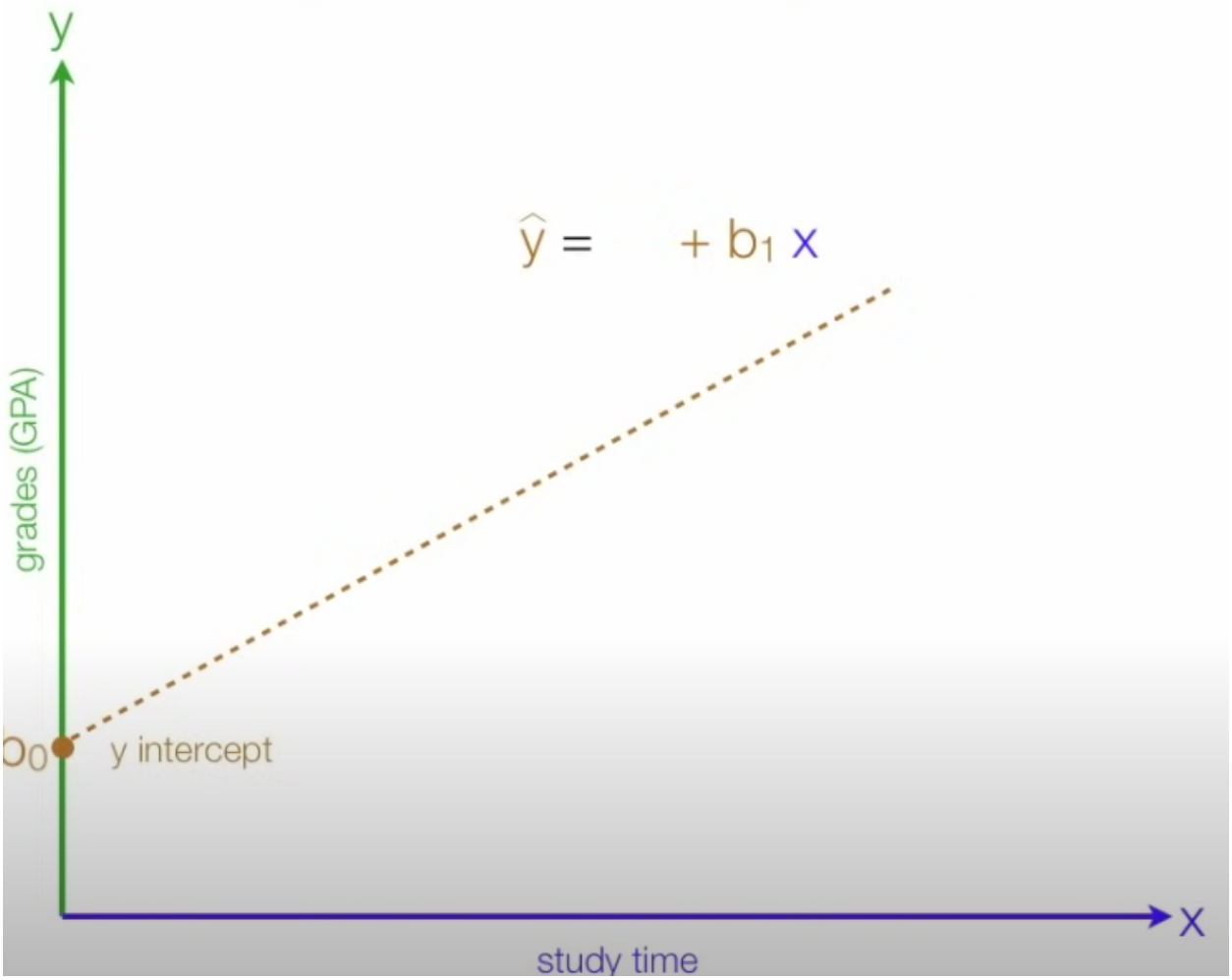
Mathematical form

In regression, we develop these equations like this:

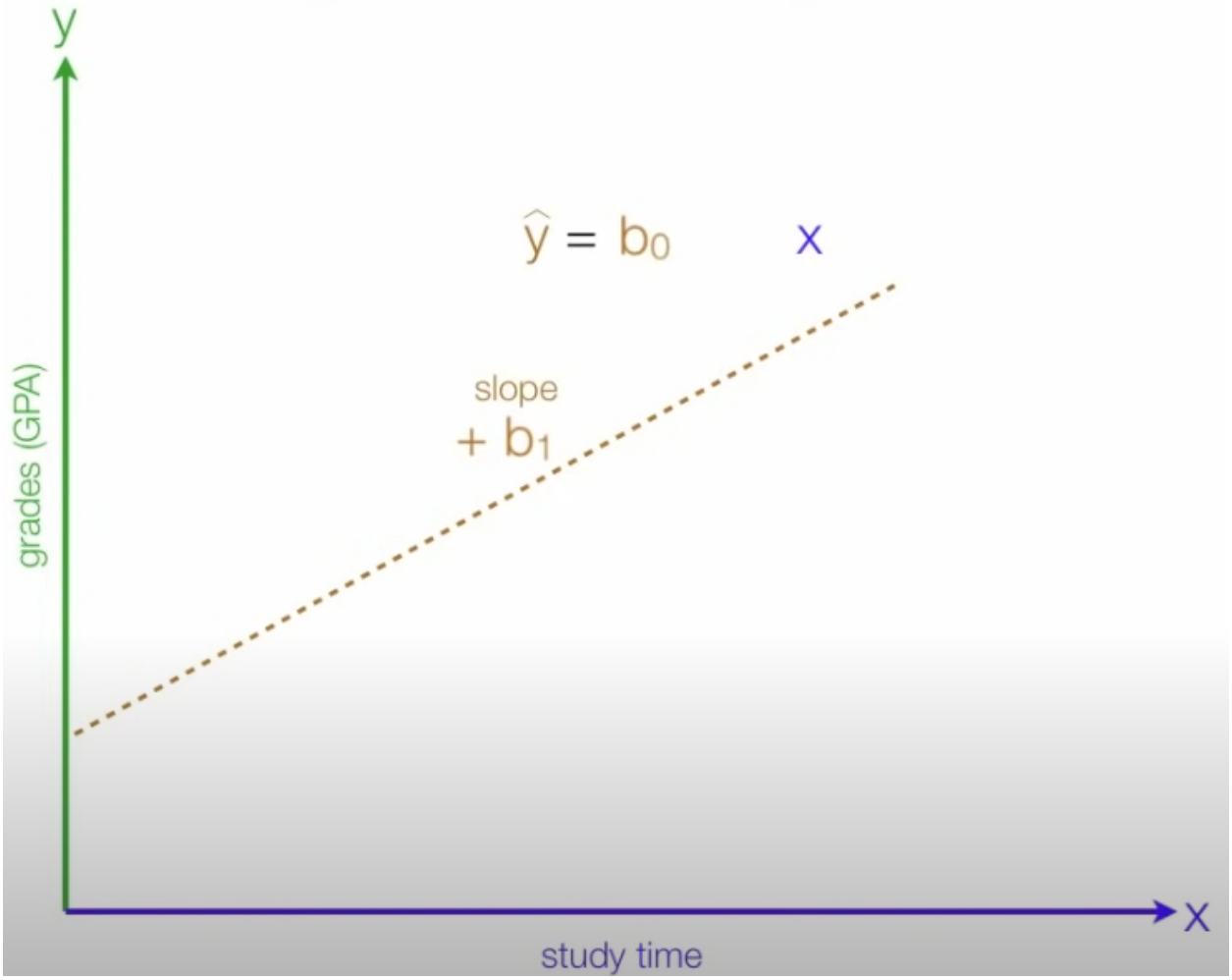
$$y_{pred} = b_0 + b_1 x$$

where x is the independent variable (e.g. study time) and y_{pred} is the estimated output values (e.g. GPA).

Here b_0 we derive mathematically which is the `y intercept`.



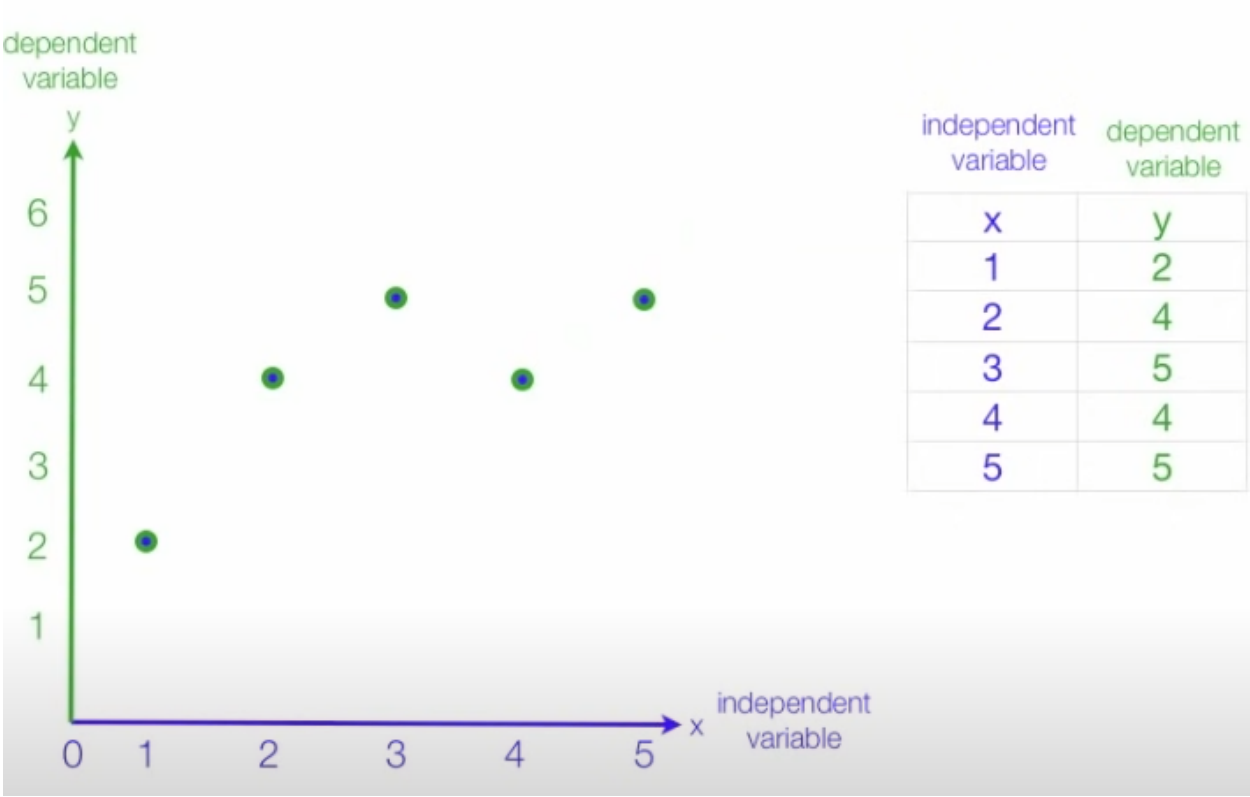
b_1 we also derive mathematically and it is the `slope` os the line.



b. For the given data (observation), how do you estimate the parameters? Use mathematical notations for your explanation. Also, talk about the specific choice of loss (objective) functions and the rationale behind it.

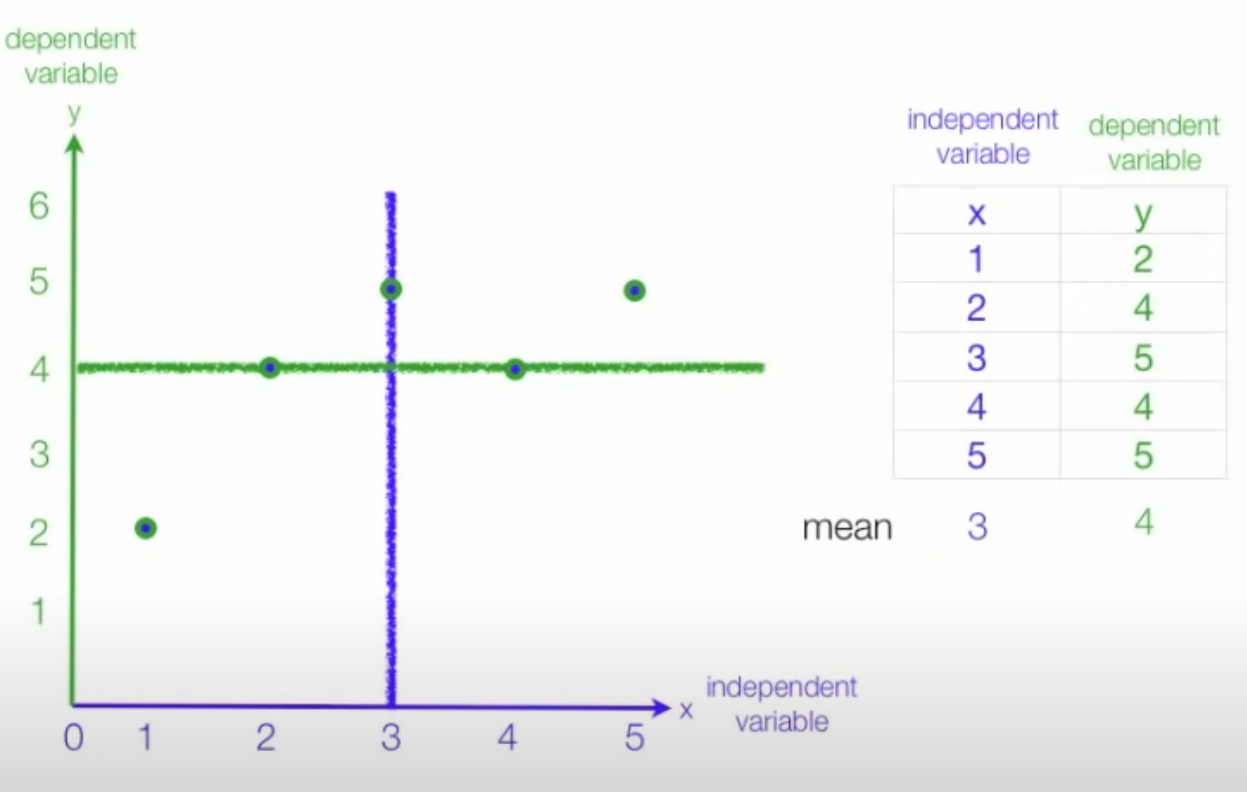
I will explain this with an example.

Suppose we have some some observations and we plot them.



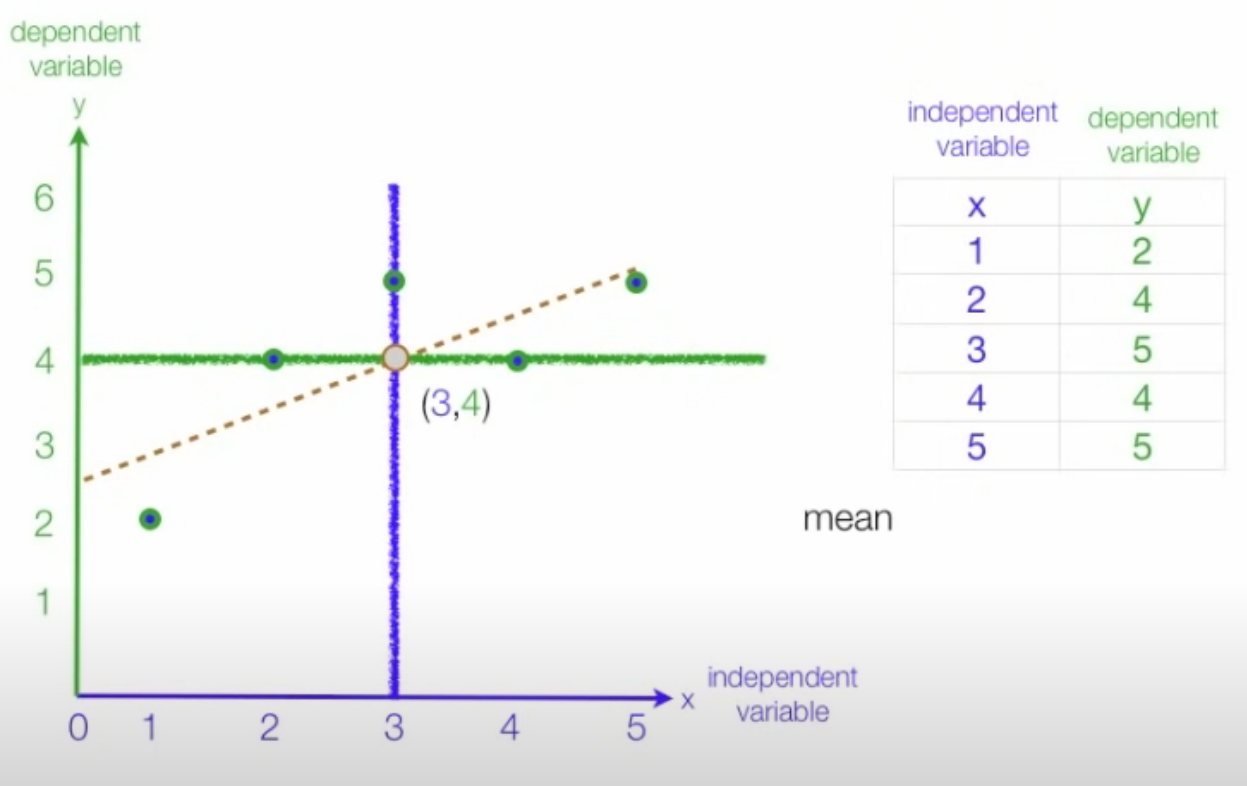
independent variable	dependent variable
x	y
1	2
2	4
3	5
4	4
5	5

The we take `mean` of x and y values and draw the lines.



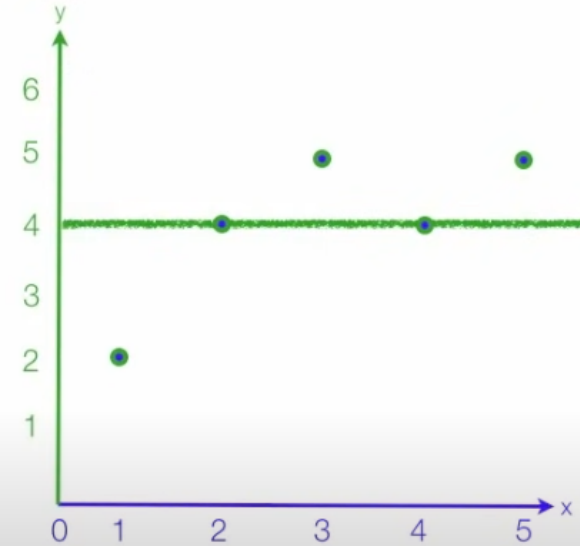
	independent variable	dependent variable
	x	y
	1	2
	2	4
	3	5
	4	4
	5	5
mean	3	4

It turns out that all the regression lines have to pass through the point where mean of the dependent and independent variable cross.



independent variable	dependent variable
x	y
1	2
2	4
3	5
4	4
5	5

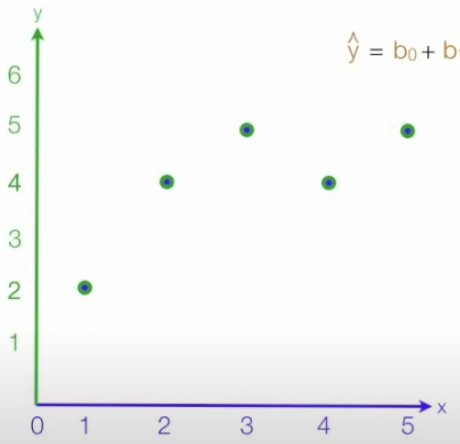
To do this, we take distance from the x-values to the mean for all obervations, and do the same thing for y-values.



x	y	x - $\bar{x}$	y - $\bar{y}$
1	2	-2	-2
2	4	-1	0
3	5	0	1
4	4	1	0
5	5	2	1
mean	3	4	

Our goal is to determine b_0 and b_1 in the equation $y_{ptred} = b_0 + b_1 x$.

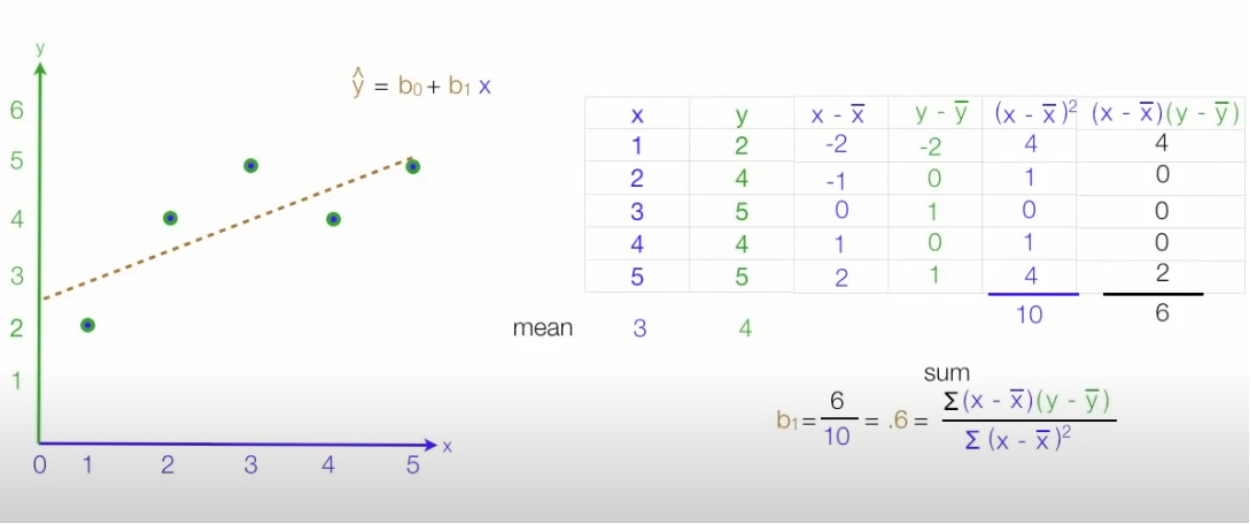
First we determine b_1 i.e. slope. We square the `x-x_mean` values. Then we make a new column and multiple `(x-x_mean)` with `(y-y_mean)`.



x	y	x - $\bar{x}$	y - $\bar{y}$	$(x - \bar{x})^2$	$(x - \bar{x})(y - \bar{y})$
1	2	-2	-2	4	4
2	4	-1	0	1	0
3	5	0	1	0	0
4	4	1	0	1	0
5	5	2	1	4	2
mean	3	4			

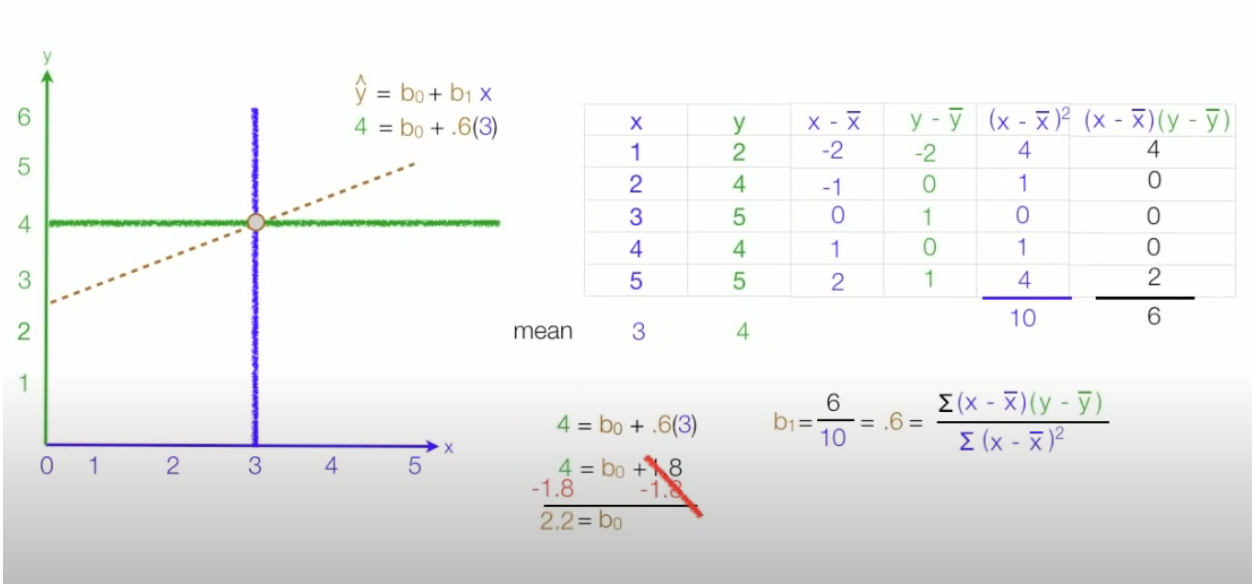
Then we calculate b_1 by:

$$b_1 = \frac{\sum(x - x_{mean})((y - y_{mean})}{\sum(x - x_{mean})^2}$$



x	y	x - x̄	y - ȳ	(x - x̄)²	(x - x̄)(y - ȳ)
1	2	-2	-2	4	4
2	4	-1	0	1	0
3	5	0	1	0	0
4	4	1	0	1	0
5	5	2	1	4	2
mean	3	4		10	6

Noe the regression lines has to cross the cross point of two mean lines. Here y-value is 4 in this particular case. Now we can plot these values into the equation and get b_0.



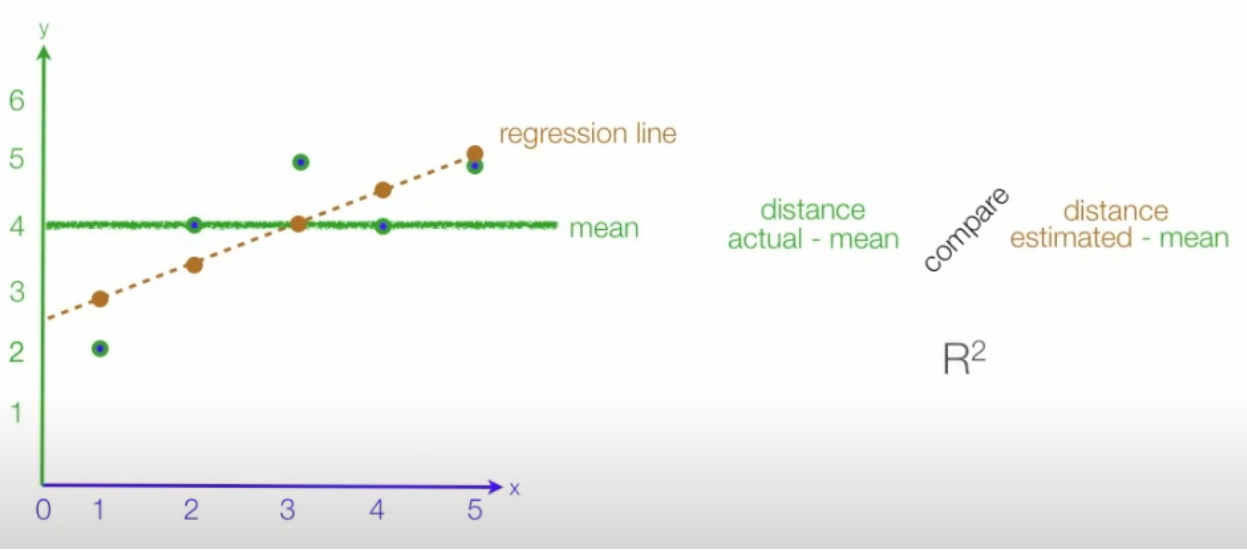
x	y	x - x̄	y - ȳ	(x - x̄)²	(x - x̄)(y - ȳ)
1	2	-2	-2	4	4
2	4	-1	0	1	0
3	5	0	1	0	0
4	4	1	0	1	0
5	5	2	1	4	2
mean	3	4		10	6

Cost Functions

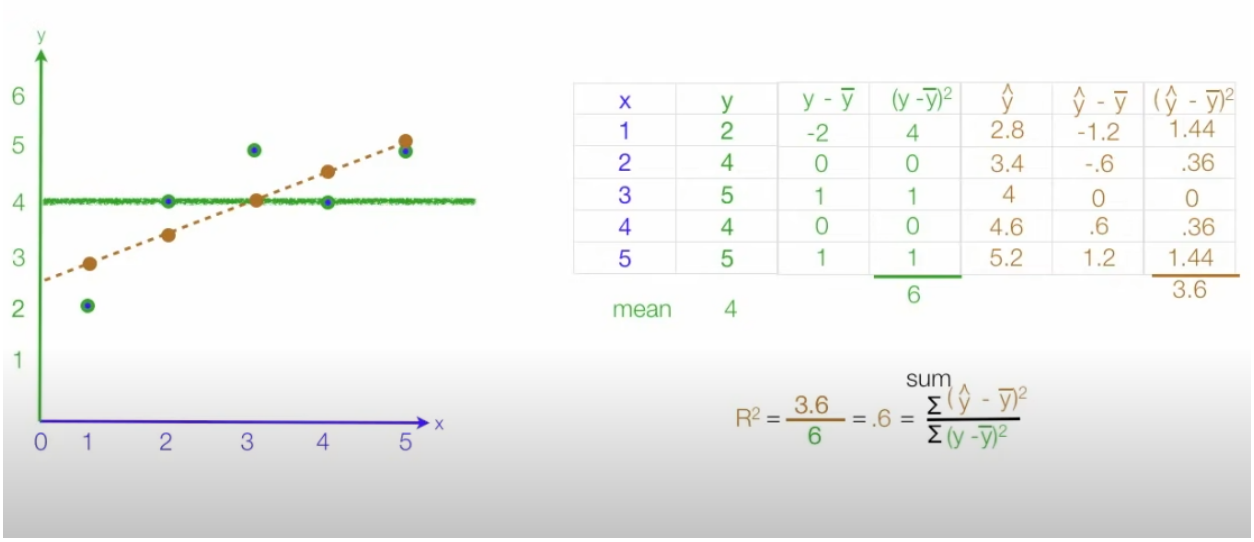
For cost function, we can use R^2 or `mean square error.`

R-squared

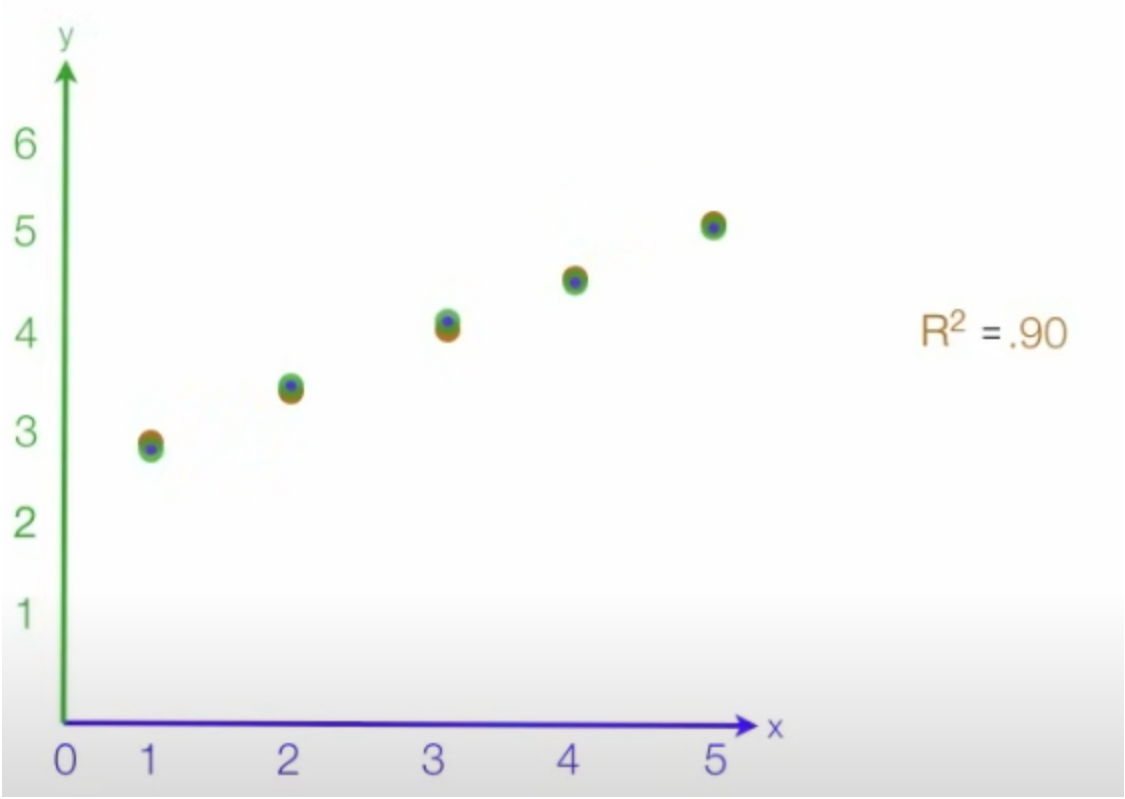
Here we compare the distance of estimated values from the mean and the distance of the actual values from the mean.

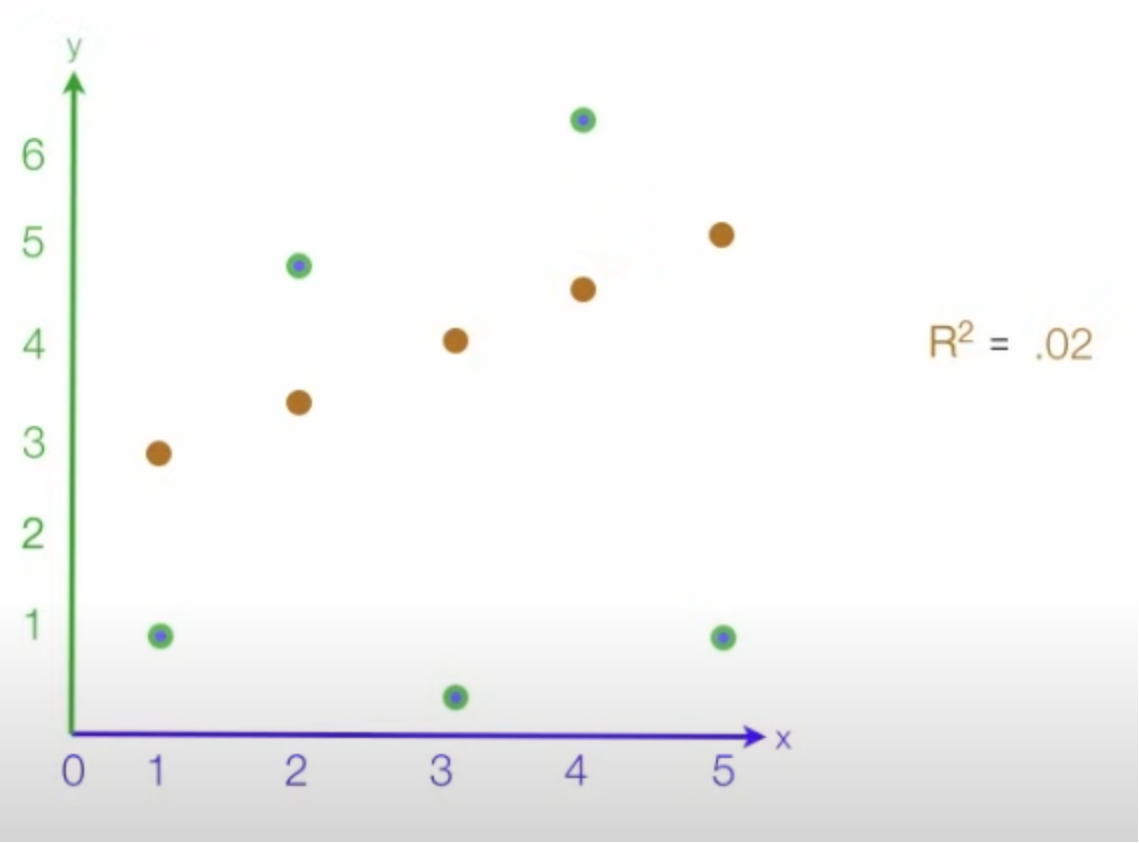


For the above example, the calculations are as follows:



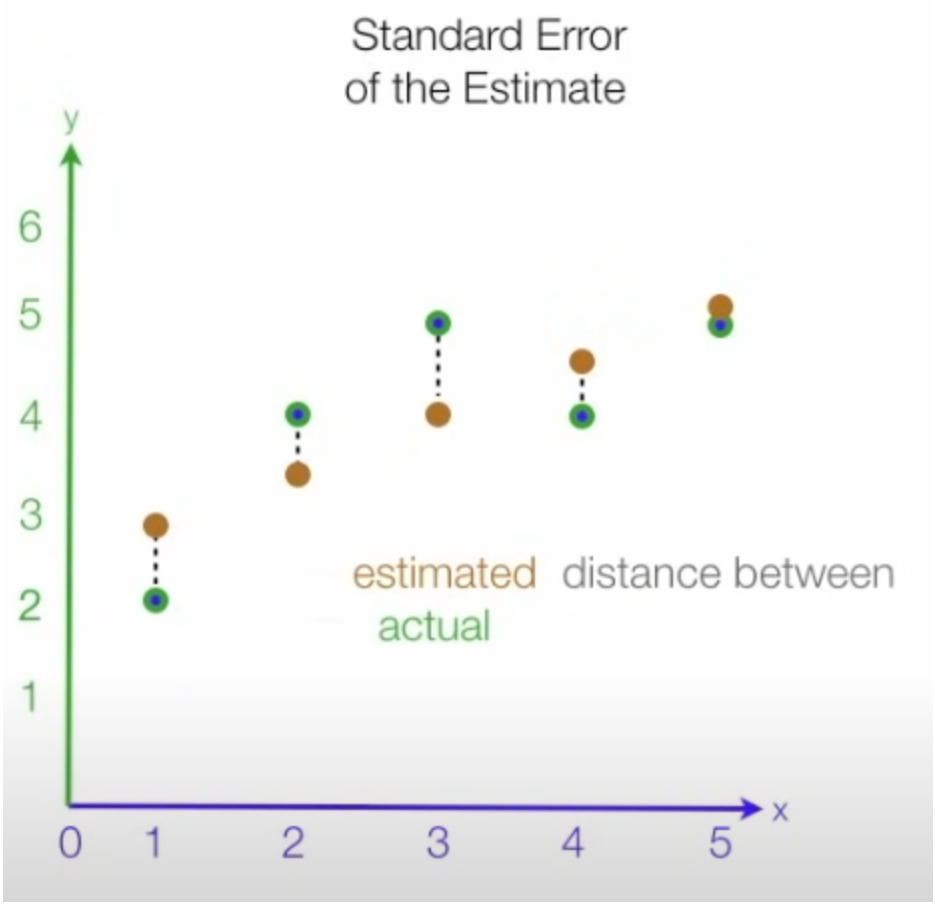
If the `actual` and `estimated` values are close, then the R^2 values is high and vice versa.



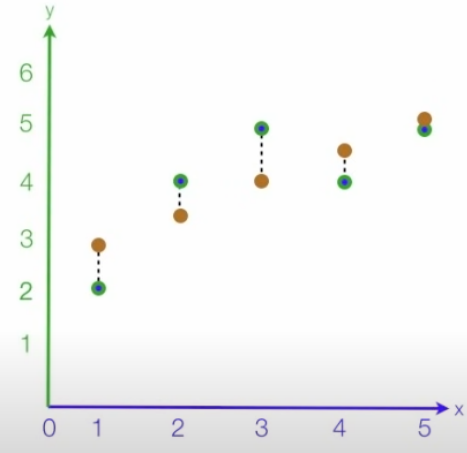


Mean Square Error

Here we calculate the distance between the `estimated` and `actual` values. These distances are called errors and we try to minimize them.

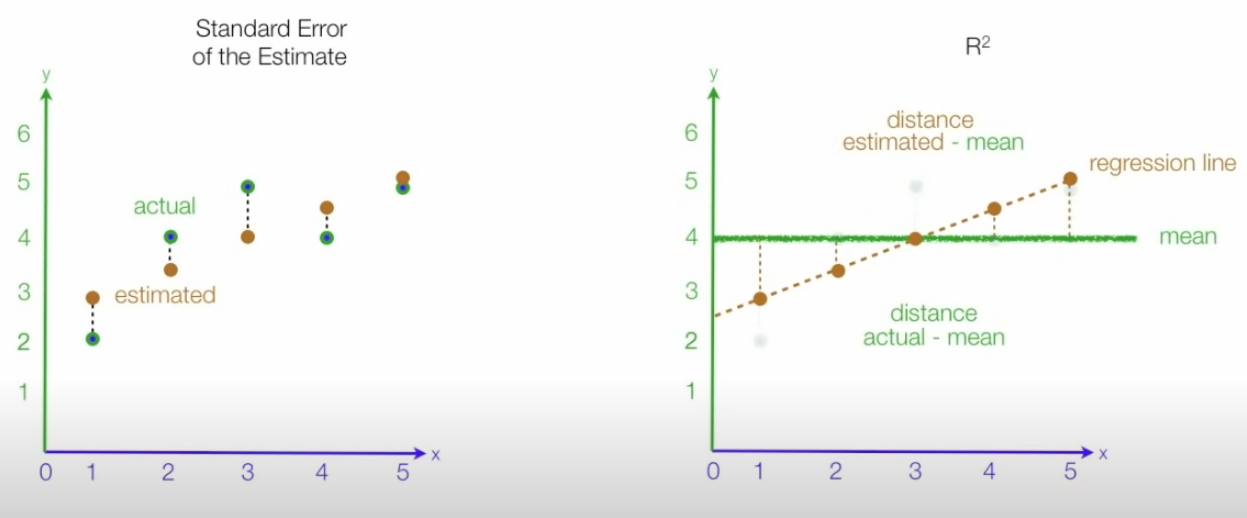


The following example illustrates some calculations.



x	y	$\hat{y}$	$\hat{y}$ - y	($\hat{y}$ - y)^2
1	2	2.8	.8	.64
2	4	3.4	-.6	.36
3	5	4	-1	1
4	4	4.6	.6	.36
5	5	5.2	.2	.04
				2.4

Standard Error of the Estimate $= \sqrt{\frac{\Sigma(\hat{y} - y)^2}{n - 2}} = \sqrt{\frac{2.4}{5-2}} = \sqrt{\frac{2.4}{3}} = \sqrt{.8} = .89$



2. [Machine Learning] Choose a machine learning model apart from a neural network, linear regression, and logistic regression. Explain the following points.

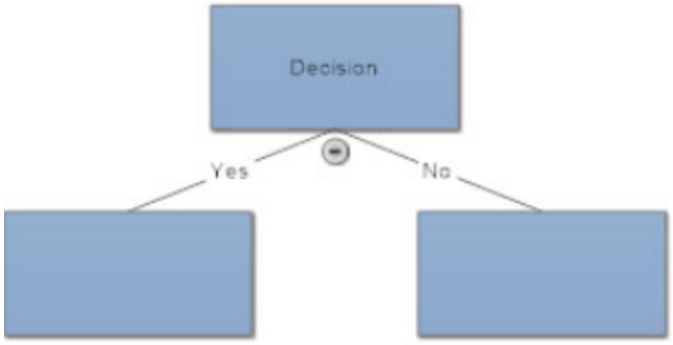
I choose **Decision Trees**.

a. What kind of problem the model can be applied to (the domain of the problem).

They are used for **classification** and **regression** tasks.

b. Structure of the model.

Decision trees have three main parts: a root node, leaf nodes and branches. The `root` node is the starting point of the tree, and both root and `leaf` nodes contain questions or criteria to be answered. `Branches` are arrows connecting nodes, showing the flow from question to answer. Each node typically has two or more nodes extending from it. For example, if the question in the first node requires a "yes" or "no" answer, there will be one leaf node for a "yes" response, and another node for "no."



c. The time complexity of the model (make reasonable assumptions as needed).

What happens in the training stage is that for each of the features (dimensions) in the dataset we'll sort the data which takes O(n log n) time following which we traverse the data points to find the right threshold which takes O(n) time. For d dimensions, total time complexity would be:

$$O(nlogn * d) + O(n * d)$$

$$which\ asymptotically\ is$$

$$O(nlogn * d)$$

d. Training method.

Decision trees can be learned from training
data. Training data will typically comprise many instances.

The decision tree learning algorithm recursively learns the tree as follows:

```
1. Assign all training instances to the root of the tree. Set
curent node to root node.

2. For each attribute, partition all data instances at the node by the value of the
attribute.

3. Compute the information gain ratio from the partitioning.

4. Identify feature that results in the greatest information
gain ratio. Set this feature to be the splitting criterion at the current node.

5. If the best information gain ratio is 0, tag the current
node as a leaf and return.

6. Partition all instances according to attribute value of the
best feature.

7. Denote each partition as a child node of the current node.

8. For each child node:

If the child node is “pure” (has instances from only one
class) tag it as a leaf and return.
If not set the child node as the current node and recurse to
step 2.
```

e. Pros and cons compared to other machine learning models.

Advantages:

- Easy to understand and interpret, perfect for visual representation. This is an example of a white box model, which closely mimics the human decision-making process.
- Can work with numerical and categorical features.
- Requires little data preprocessing: no need for one-hot encoding, dummy variables, and so on.
- Non-parametric model: no assumptions about the shape of data.

- Fast for inference.
- Feature selection happens automatically: unimportant features will not influence the result. The presence of features that depend on each other (multicollinearity) also doesn't affect the quality.

Disadvantages:

- Tends to overfit
- Inadequate prediction powers
- If dataset is not balanced this can cause a bias.

---

1. [Open Question] You are building an API service that can classify images. You've gathered about 1M images from the web and labeled 100k images. Using the labeled images, you've trained your ResNet model according to the training method described in the paper, but you were unable to meet the accuracy requirement of the service. What would you do to improve the accuracy of the model? (However, due to the response speed of the service, it is not possible to increase the computation of the model.)

a. Make reasonable assumptions as much as required.

Since we cannot change the model architecture due to computational restraint, we can do one of the followings:

Use More Data

A deep learning model is as good as the amount of data it gets. One of the easiest ways to increase validation accuracy is to add more data.

Change Image Size

When you preprocess your images for training and evaluation, there is a lot of experimentation you can do with regards to the image size.

If you choose an image size that is too small, your model will not be able to pick up on the distinctive features that help with image recognition.

Conversely, if your images are too big, it increases the computational resources required by your computer and/or your model might not be sophisticated enough to process them.

Common image sizes include 64x64, 128x128, 28x28 (MNIST), and 224x224 (VGG-16).

Increase epochs

Epochs are basically how many times you pass the entire dataset through the neural network. Incrementally train your model with more epochs with intervals of +25, +100, and so on.

Decrease color channels

The more complex the colour channels are, the more complex the dataset is and the longer it will take to train the model.

If colour is not such a significant factor in your model, you can go ahead and convert your colour images to grayscale.

Transfer Learning

This is probably one of the most useful learning technique. Instead fo rtaining the model from scratch, use a `pretrained model`.

Pre-trained models are state-of-the-art deep learning models that were trained on millions and millions of samples, and often for months. These models have an astonishingly huge capability of detecting nuances in different images.

Nowadays you can also use some recent advanced techniques like meta-learning and self-supervised learning to increase your models performance.