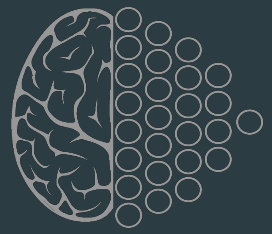
Lecture 4

INTRODUCTION AND APPLICATIONS OF MACHINE LEARNING



Contacts

Instructor

- Shreyas Bhat - 9082080984
- S I Harini - 7021247073
- Rishav Mukherji - 9920741703

Mentor

- Hardik Shah - 9427925103

Roadmap

- Machine Learning terms
  - Machine learning pipeline
  - Representing data
  - Supervised learning
  - Unsupervised and semi supervised learning
    - K Means Clustering
- Linear regression
  - Mean squared error
  - Gradient descent and Gradient update
  - Multiple linear regression
  - Bayesian linear regression

Machine Learning Pipeline

- Data generation/collection
- Data Cleaning/Feature Engineering
- Applying Algorithms according to pattern
- Deployment

Supervised Learning

- Learning from labelled data(training)
- Goal is to generalize predictions on future data
- Divided into two categories of algorithms -
  - Classification: A problem where the output variable is a category. Such as tumour malignant vs benign
  - Regression: The output is a continuous outcome and we have to predict it from out train data

Training

- All input data have associated labels
- Label can be continuous value or discrete value(apple or not)
- Input data are features - Color of the fruit, size of the fruit, sweetness of the fruit etc.
- Every input is associated with some features and labels
- At training, you would be given both x and y values (x train and y train)

Representation of data

```
In [7]: df = pd.read_csv("train.csv")
        df.head(25)
```

Out[7]:

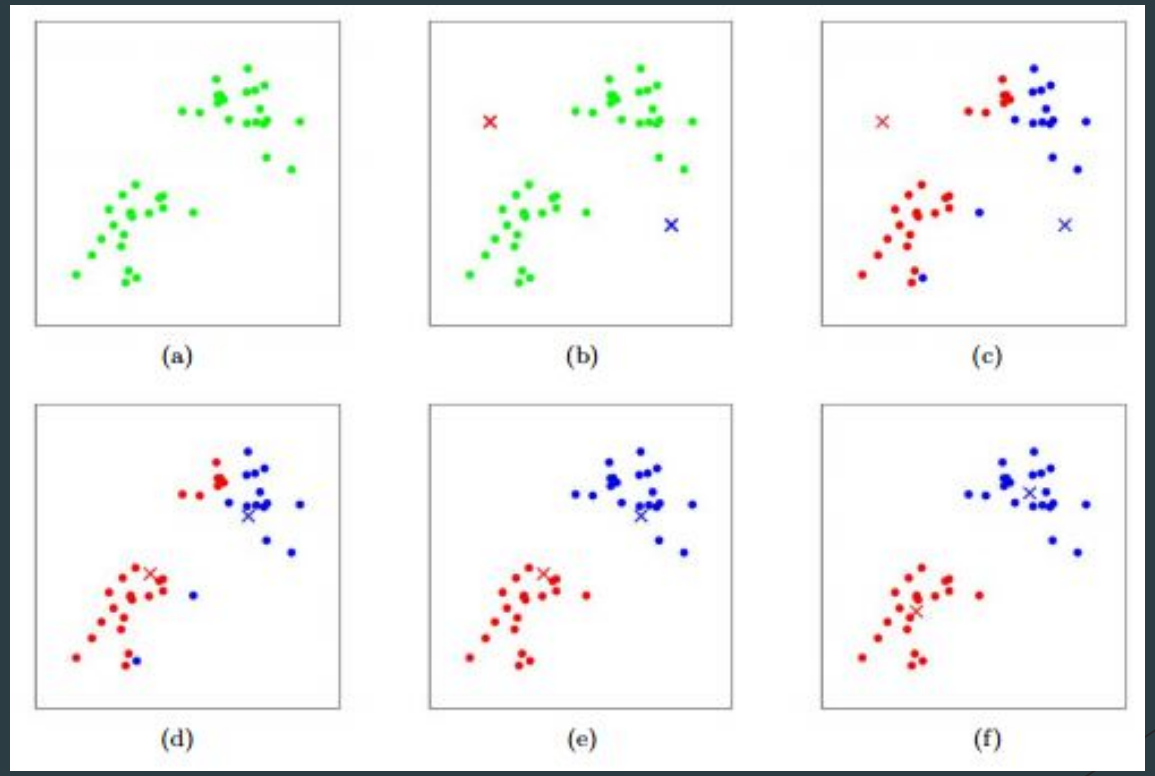
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S

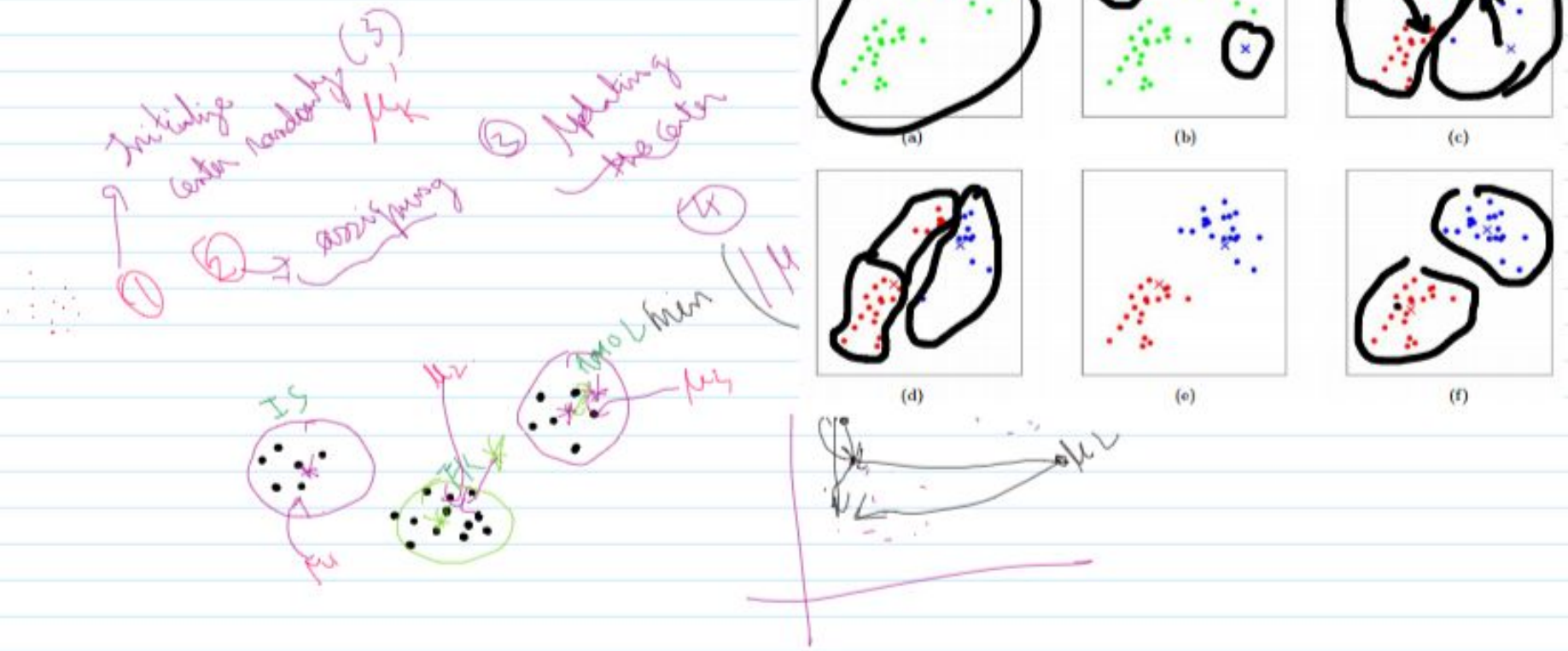
Unsupervised Learning

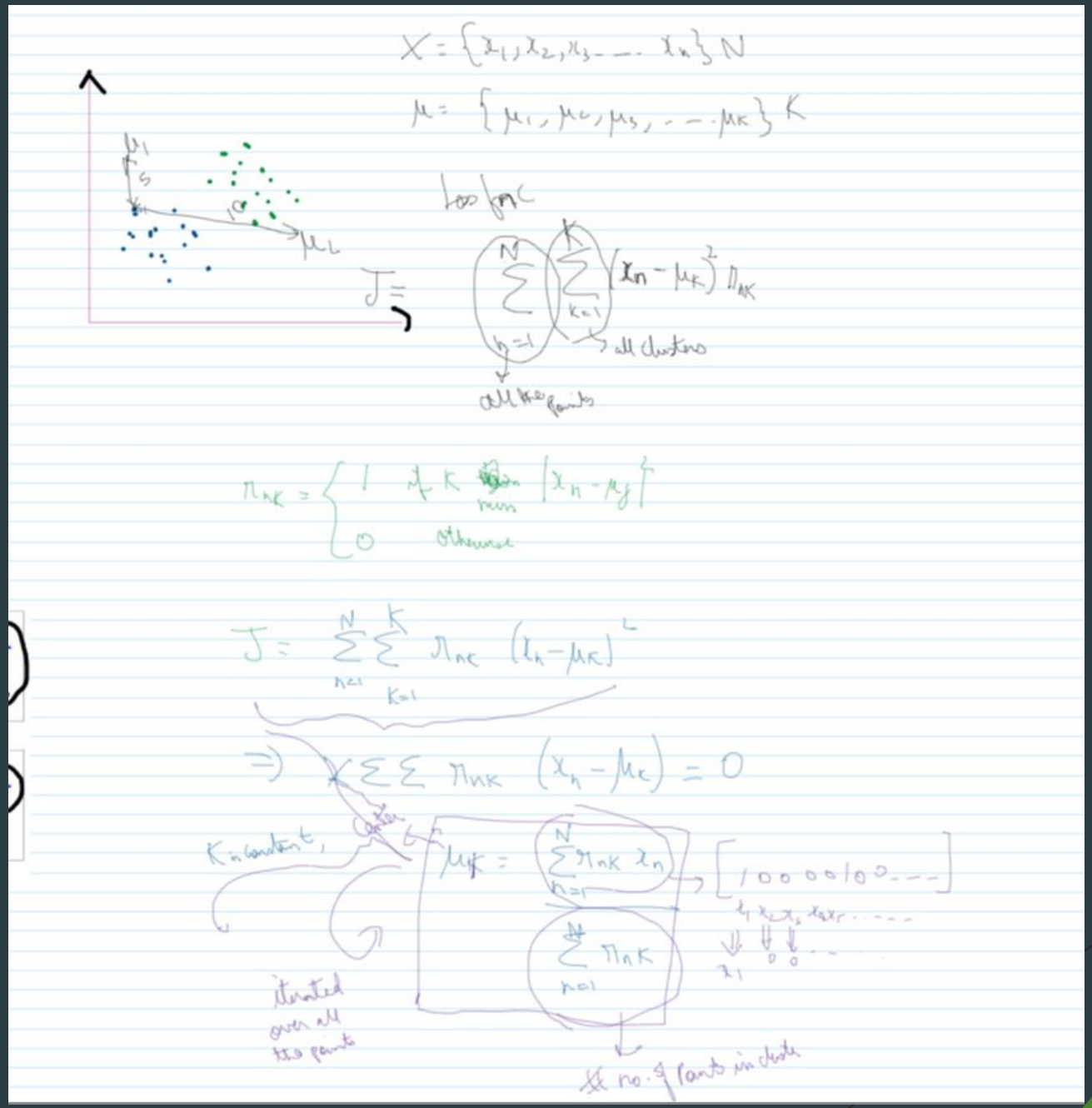
- No supervision ie no labelled data
- The algorithm has to find the pattern on its own from the structure of the data - groups, clusters, similar points together
- Only the x values and y values is not known
- Doesn’t require human supervision for labelling the data

K-means clustering

- Common unsupervised clustering technique
- K-means is an example of Hard-clustering ie every point belongs to only one cluster
- K = number of clusters
- Sensitive to initialization




Initialize center randomly (3)
assigning
Updating the center
(a)
(b)
(c)
(d)
(e)
(f)


$X = \{x_1, x_2, x_3 \ldots x_n\}\ N$
$\mu = \{\mu_1, \mu_2, \mu_3, \ldots \mu_K\}\ K$
$J =$
$\sum_{n=1}^{N} \sum_{k=1}^{K} (x_n - \mu_k)^2\, \pi_{nk}$
all clusters
all the points
$\pi_{nk} = \begin{cases} 1 & \text{if } k \text{ min} \|x_n - \mu_j\|^2 \\ 0 & \text{otherwise} \end{cases}$
$J = \sum_{n=1}^{N} \sum_{k=1}^{K} \pi_{nk} (x_n - \mu_k)^2$
$\Rightarrow \sum\sum \pi_{nk} (x_n - \mu_k) = 0$
K is constant
$\mu_j = \frac{\sum_{n=1}^{N} \pi_{nk} x_n}{\sum_{n=1}^{N} \pi_{nk}}$
[1 0 0 0 0 1 0 0 ...]
iterated over all the points
no. of points in cluster

Semi-supervised learning

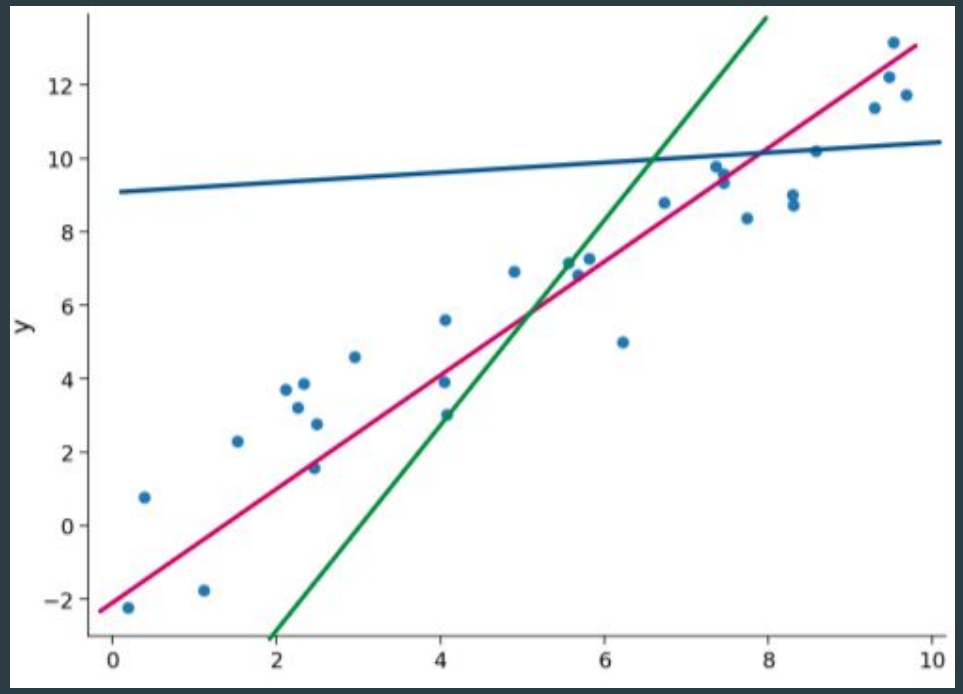
- The algorithm is trained on a combination of labeled and unlabeled data.
- Usually we use a small amount of labeled data and a very large amount of unlabeled data
- The basic procedure is:
  - We cluster similar kind of data using an unsupervised algorithm
  - Then use labeled data to label some portion of the unlabeled data
- So when will this fail?
- Decreasing order of accuracy

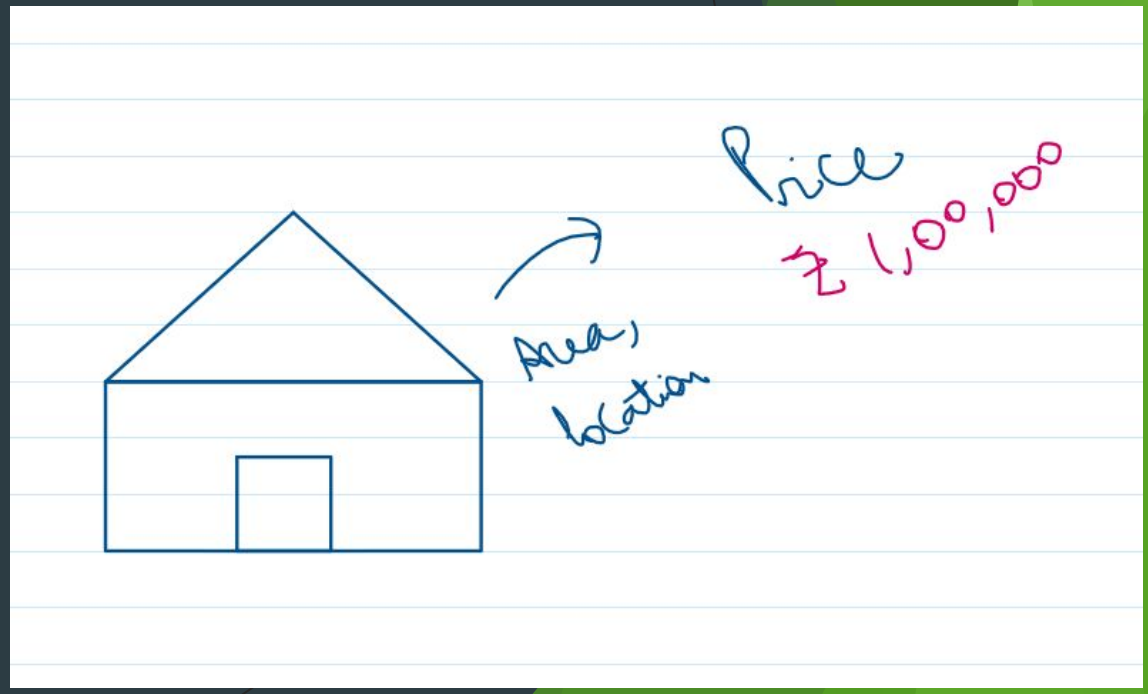
Application of semi-supervised learning

- **Protein Sequence Classification:** Since DNA strands are typically very large in size, the rise of Semi-Supervised learning has been imminent in this field.
- https://ai.googleblog.com/2016/10/graph-powered-machine-learning-at-google.html

Linear Regression Algorithms

- Regression is predicting continuous values
- Useful for finding relationship between two variables
- The core idea is to find a line that best fits the data





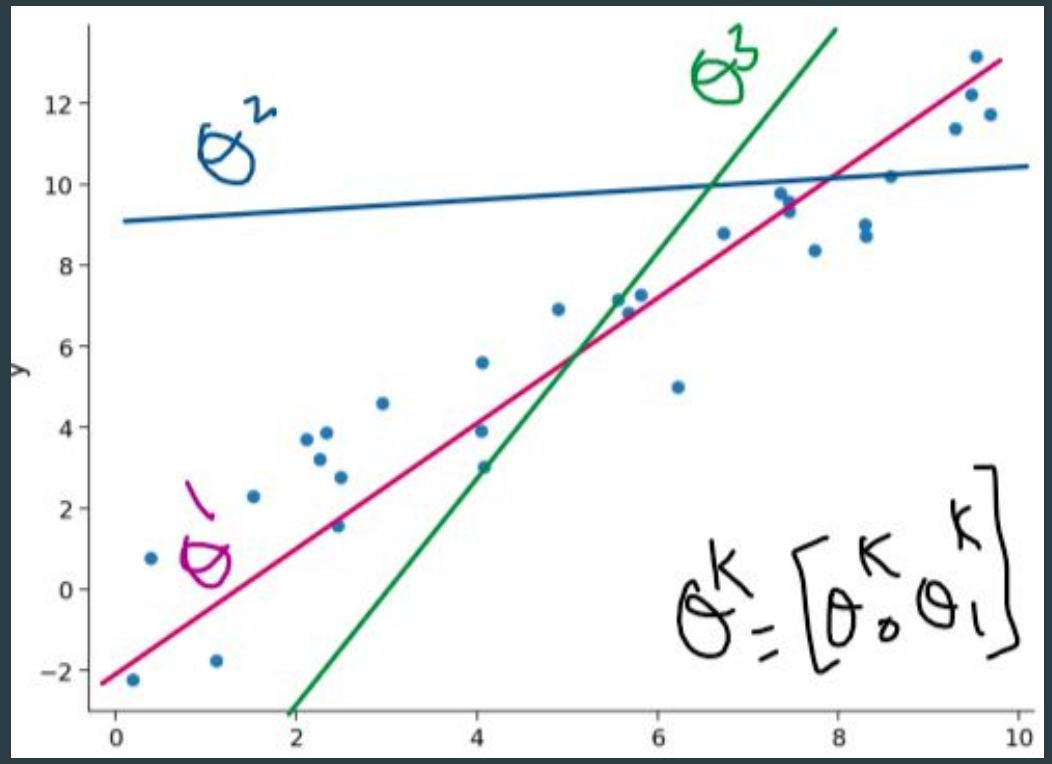
Time Spent	Score
2	2
4	10
5	12
8	15

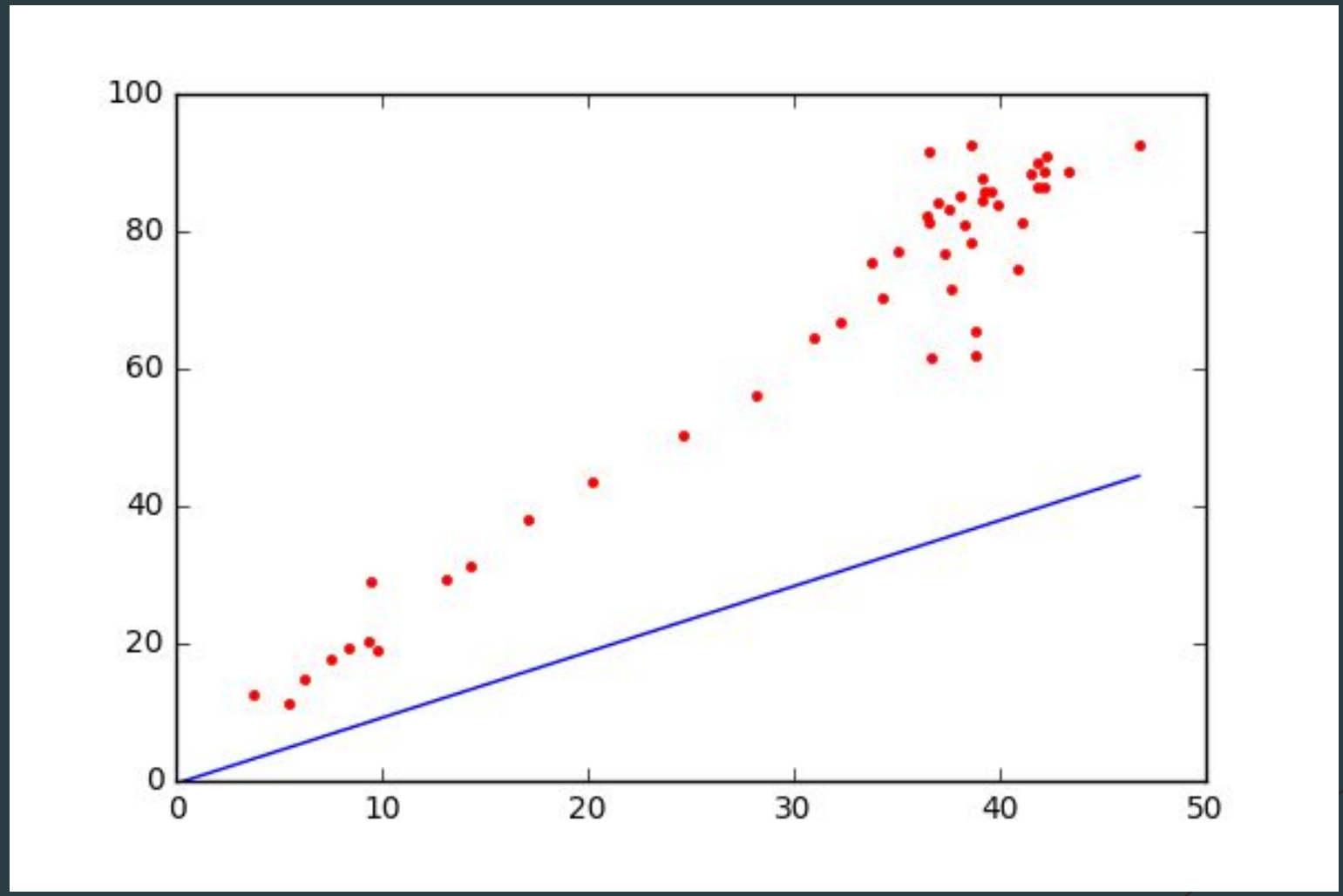
1	?
10	?

$$y = \theta_1 \times x + \theta_0$$

How good is our line?

- A good theta is the one that reduces our error
- So, we have to reduce our error on training data

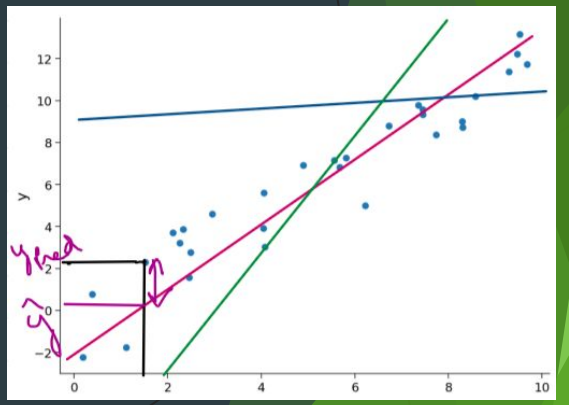


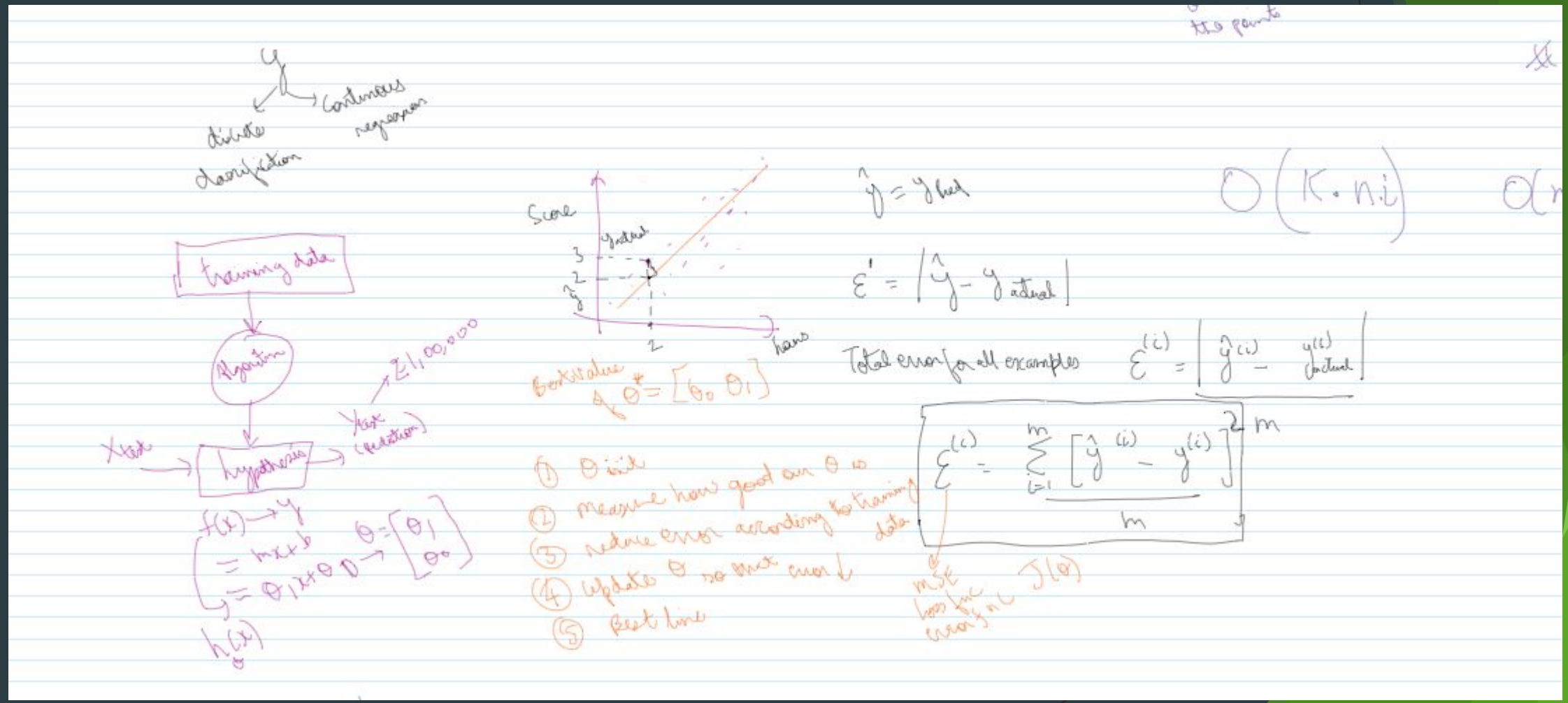

100
80
60
40
20
0
0
10
20
30
40
50

Mean squared error

The **mean squared error** (MSE) tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them. The squaring is necessary to remove any negative signs. It also gives more weight to larger differences. It’s called the **mean** squared error as you’re finding the average of a set of errors. The lower the MSE, the better the forecast.

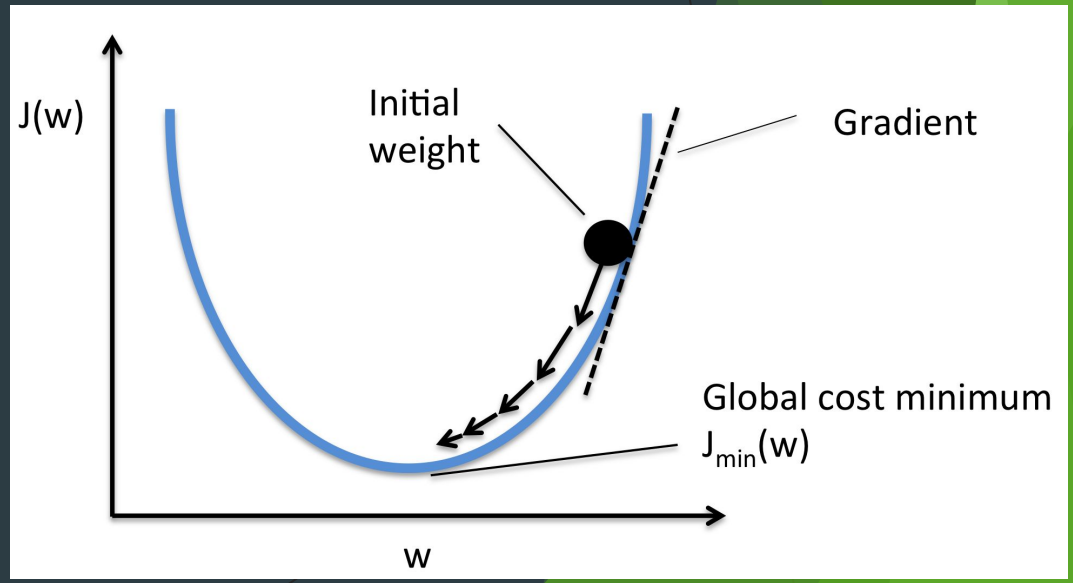
$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$



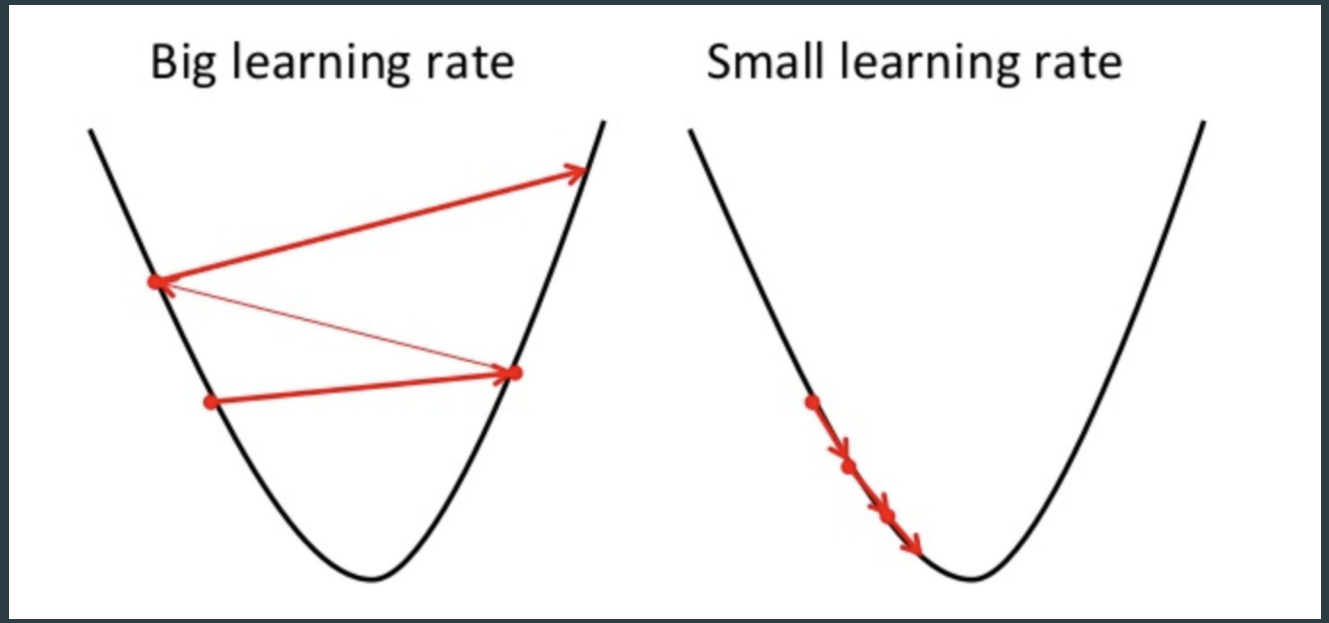

y
discrete classification
continuous regression
training data
Algorithm
hypothesis
x_{test}
y_{test} (prediction)
$h(x)$
$f(x) \rightarrow y$
$= mx+b$
$= \theta_1 x + \theta_0$
$\theta = \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix}$
Score
y_{actual}
Hours
Best value $\theta^* = [\theta_0 \ \theta_1]$
1) θ init
2) measure how good our θ is
3) reduce error according to training data
4) update θ to best error
5) Best line
$\hat{y} = y_{hat}$
$\varepsilon' = |\hat{y} - y_{actual}|$
Total error for all examples
$\varepsilon^{(i)} = |\hat{y}^{(i)} - y^{(i)}_{actual}|$
$\varepsilon^{(i)} = \sum_{i=1}^{m} [\hat{y}^{(i)} - y^{(i)}]^2 / m$
$\varepsilon^{(i)} = \sum_{i=1}^{m} [\hat{y}^{(i)} - y^{(i)}] / m$
mse loss func error func
$J(\theta)$
$O(K \cdot n \cdot i)$

Gradient Descent

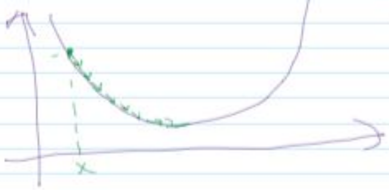
- Optimization method to find minima in a function (here reduce the cost function MSE)
- It is an iterative process
- Start at any point and move towards the minima
- This depends on
  - Step size (η)
  - direction(determined by the negative of the gradient)



Gradient update



Gradient descent algorithm



$y = f(x)$

$f(x) = (x-5)^2$

$X = X - \eta \frac{\partial f(x)}{\partial x}$ = 0

η: constant (learning rate)

$\frac{\partial f(x)}{\partial x}$: gradient / slope

1. θ init
2. $J_\theta = \frac{1}{m} \sum_{i=1}^{m} [y^{(i)} - y^{(i)}_{actual}]^2$
3. update θ using GD

$J(\theta) = \frac{1}{m} \sum [\theta_0 + \theta_1 x]$

$\theta = \theta - \eta \nabla J(\theta)$ → θ_0, θ_1

⇒ $\theta_0 = \theta_0 - \eta \frac{\partial J(\theta)}{\partial \theta_0}$ $\theta_1 = \theta_1 - \eta \frac{\partial J(\theta)}{\partial \theta_1}$

$J(\theta) = \frac{1}{m} \sum_{i=0}^{m} [(\theta_0 + \theta_1 x^{(i)}) - y^{(i)}_{actual}]^2$

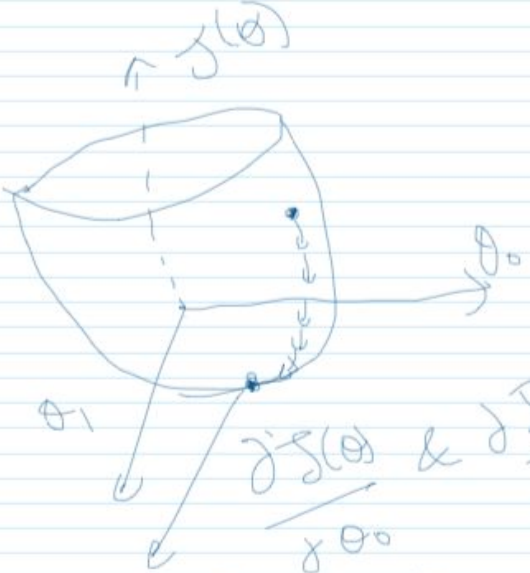
1. $\frac{\partial J(\theta)}{\partial \theta_0} = \frac{2}{m} \sum_{i=0}^{m} [\theta_0 + \theta_1 x^{(i)} - y^{(i)}_{actual}] \cdot 1$

2. $\frac{\partial J(\theta)}{\partial \theta_1} = \frac{2}{m} \sum_{i=0}^{m} [\theta_0 + \theta_1 x^{(i)} - y^{(i)}_{actual}] \cdot x^{(i)}$

finally,

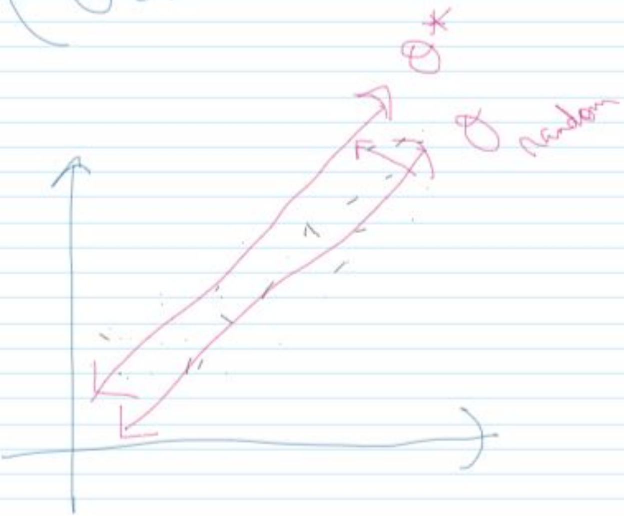
$\theta_0 = \theta_0 - \eta \frac{1}{m} \sum_{i=1}^{m} [\hat{y}^{(i)} - y^{(i)}_{actual}] 1$

$\theta_1 = \theta_1 - \eta \frac{1}{m} \sum_{i=1}^{m} [\hat{y}^{(i)} - y^{(i)}_{actual}] x^{(i)}$



$\frac{\partial J(\theta)}{\partial \theta_0}$ & $\frac{\partial J(\theta)}{\partial \theta_1}$ =

$(\theta_0, \theta_1) \rightarrow \theta^*$



THANK YOU!