# Machine Learning

## Basic

### What is Machine learning?

⇒ Machine learning is a branch of artificial intelligence and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

⇒ Machine learn just like human being. Like a child grow up, and observe his/her daily life, watch other and learn new things everyday. And in adult age he/she can do many task with better accuracy than the child age.

Machine also learn by observing data. The more data it can observe, the more accuracy it can produce.

❊ Why we use machine learning model?

⇒ Let's say, we trying to guess or predict the rent of a room. In general, $\underbrace{\text{rent of a room}}_{\text{Target}}$ depends on

- size
- location
- floor

} feature

Target : What we want to find out or predict. In this case rent of a room is our target. And it is dependent variable.

Feature: Variables at which our target dependent. features are independent variables.

Let's consider a dummy example:

| | Size | location | floor | rent cost |
|---|---|---|---|---|
| 1 | 1500 | Banani | 7 | 50,000 |
| 2 | 2200 | Tongi | 4 | 55000 |
| 3 | 1200 | Banani | 2 | 30,000 |
| 5 | 1500 | Dhanmondi | 5 | 40,000 |

Here,

(i) if we set a single rule on size, like size increase, rent cost will increase. But we can see that same size has different rent cost. Also the increasing rate is not same for all. So, we can't use this rule.

(ii) If we set a single rule on location, like high developed location have better price. But same location have different price. So, not possible.

(iii) If we set a rule on floor, like ~~ki~~ appen floor has better price. But here we can see that,

2 → 4 ⇒ increase
4 → 5 ⇒ decrease

So, not possible.

⇒ One possible way is to ~~ste~~ set rule combining every feature. Like

```
if (location == Banani && size == 1500 &&
        floor == 7):
    rent cost = 50,000
else if ( . . .
    - - -
    - - -
```

⇒ But here is another problem exist. Like if we want to predict rent of a new location, we can't get answer

Example:

location = Uttara
size = 1800
floor = 3

There are not set on the if-else. So our code will not predict the rent cost.

⇒ So, when we cannot set the rules, we use machine learning to predict the target.

⇒ Machine learning model, finds the general solution to predict the target.

# Lecture - 2

## Types of Machine Learning

There are 4 types:

1. Supervised:
   When we use a dataset with label, then it called as supervised problem. Label included means, we exactly know the target value of each sample in the dataset.

2. Unsupervised:
   - Label is not included
   - machine learning model distinguish these data into different classes according to the features similarity.

3. Semi-Supervised:

4. Reinforcement Learning

Supervised is divided into two type problem.

(i) Regression Model

- used for predict continuous data.
- when target is unknown,

(ii) Clasification Model

- target is known
- used for categorical data.
  - or discreate data

## Lecture-3

### Linear Regression-1

$$y = f(x) = mx + c$$

dependent variable
- our target

independent Variable
- feature

Let's consider the rent cost of room problem with one feature,

| Size | Rent Cost |
|---|---|
| 1200 | 25,000 |
| 1500 | 30,000 |
| 1800 | 35,000 |
| 2000 | ? |

- 1st Guess the function

- $$\underbrace{\text{Rent Cost} = 20 \times \text{Size}}$$

⇒ It's totally guess and random function.
- it is known as hypothesis.

- Now we can guess the rent cost using this function. But is it good enough?

⊛ Now we need to check the rent cost predicted from this function, is it equal to the actual rent cost? We need to check using the sample ~~rat~~ data.

| Size | Rent Cost | Rent Cost Predicted | Difference |
|---|---|---|---|
| 1200 | 25000 | 24,000 | 1000 |
| 1500 | 30,000 | 30,000 | 0 |
| 1800 | 35,000 | 36,000 | 1000 |

↳ These are lose

$$\therefore \text{Loss} = \sum \left| \text{Actual} - \text{Predicted} \right|$$

$$= 2000$$

⇒ We need to reduce the loss as much posible as we can.

| | Sample | each loss | $\Sigma L$ |
|---|---|---|---|
| model-1 | 3 | 1000 | 3000 |
| model-2 | 1000 | 5 | 5000 |

which is better?

⇒ for single sample loss, model-2 is better than model-1.

⇒ But if we sum up all loss for a dataset then loss will increase as we have more data in our datasets.

⇒ So, we need to average the loss.

$$\text{model-1} \Rightarrow \frac{3000}{3} = 1000$$

$$\text{model-2} \Rightarrow \frac{\Sigma L}{\text{total sample}} = \frac{5000}{1000} = 5$$

⊛ In our model,

$$\text{average loss} = \frac{\cancel{3000}\ 2000}{3} = 666.67$$

We need to reduce as much as possible.

- This known as state of the art.

⇒ Latest and lowest loss.

# Lecture-4

## Linear Regression-2

⊛ When a function is Linear?

⇒ when the power of independent variable is exactly 1, then the function is Linear.

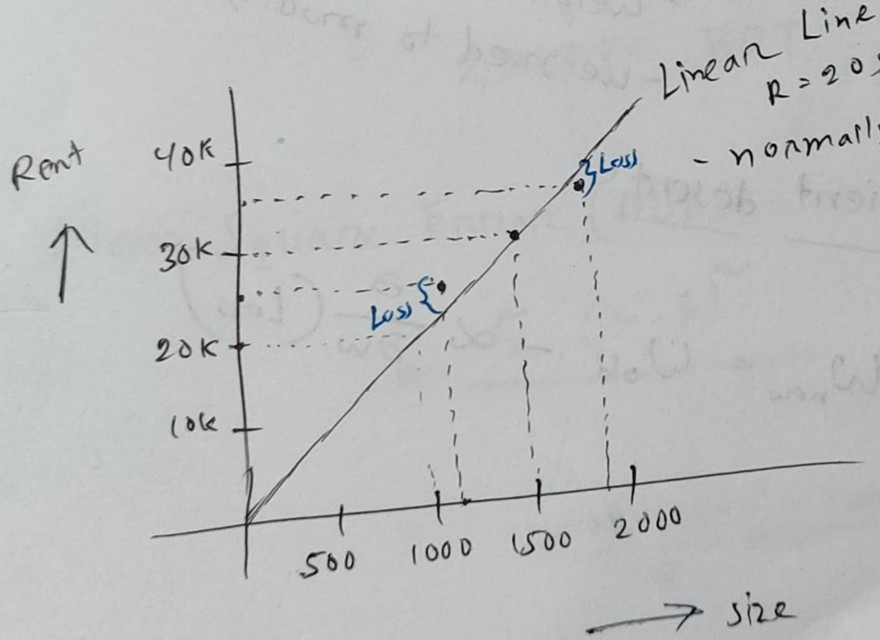
$y = 2x+3$ ✓

$y = 2x^2+3$ ×

$y = 2x^{1/2}+3$ ×

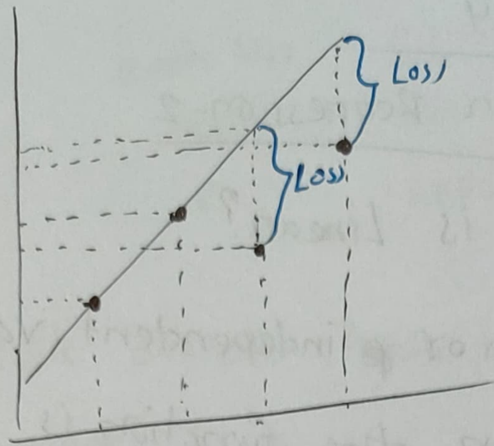
Our model was,

$R = 20s$ ⇒ Provide a linear graph.




- normally drawn as best fit line.

⇒




✱ For minimize the loss ~~fun~~, we need to modify the hypothesis.

feature

$$Y = ws$$

target

weight

- we need to modify this

We can use

- gradient desent!

$$W_{new} = W_{old} - \alpha \frac{\partial}{\partial w}(\text{Loss})$$

Lecture-5

## Regression Evaluation Metrics

| Size | actual | Predicted |
|---|---|---|
| 1000 | 20,000 | 21000 |
| 1200 | 25,000 | 27000 |
| 1,800 | 35000 | 31000 |

### Mean absolute error (MAE):

$$MAE = \frac{1}{N} \left| y - \overset{\text{Predicted}}{\hat{y}} \right|$$

$$= \frac{\Sigma (|y - \hat{y}|)}{N}$$

$$= \frac{1000 + 2000 + 4000}{3}$$

$$= 2333.34 \text{ BDT}$$

### Mean Square Error (MSE):

$$= \frac{\Sigma (y - \hat{y})^2}{N}$$

$$= \frac{1000^2 + 2000^2 + 4000^2}{3}$$

$$= 70,00,000 \text{ BDT}^2$$

Root Mean Square Error (RMSE):

$$= \sqrt{MSE}$$

$$= \sqrt{\frac{\Sigma (y-\hat{y})^2}{N}}$$

$$= 2,645.75 \text{ BDT}$$

Co-effecient of Determination:

$R^2 \Rightarrow 0 \leq R^2 \leq 1$

Best fit

Not fit

$$R^2 = 1 - \frac{\Sigma (y-\hat{y})^2}{\Sigma (y-\bar{y})^2}$$

average = ~~2666~~ 26,666.67

$$= 1 - \frac{1000^2 + 2000^2 + 4000^2}{11666666.67}$$

$$= 1 - 0.18$$

$$= 0.82 \text{ (Close to 1)}$$

# Lecture-6

## Classification Problem

| Weather | Outlook | Play golf or not |
|---|---|---|
| hot | sunny | Yes |
| cold | Sunny | Yes |
| cold | Rainy | No |
| hot | sunny | Yes |
| cold | Sunny | Yes |

⊛ **ZeroR Classifier:**

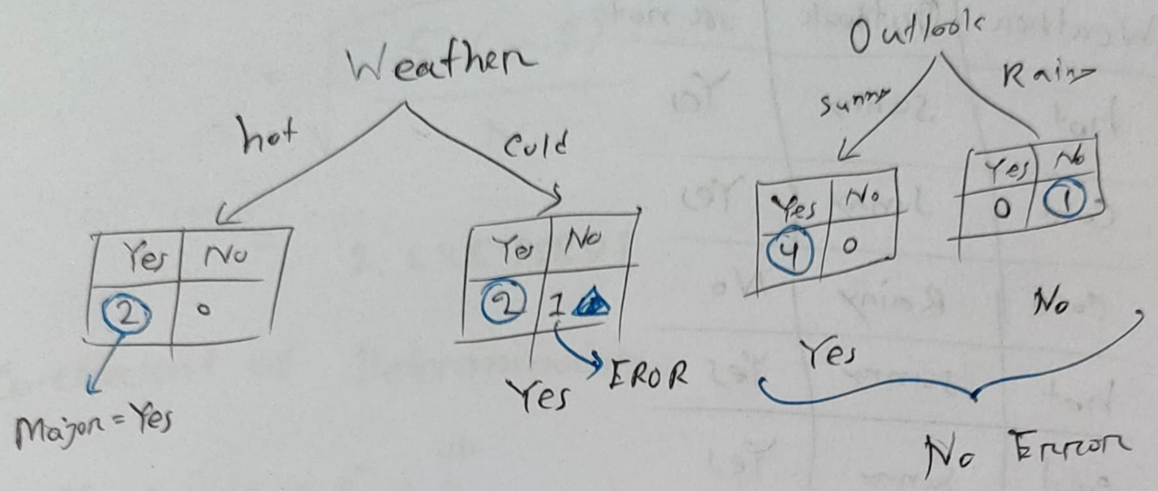
- Provide Predicted value by counting the majority.

⇒ here Yes - 4, N - 1 } Majority = Yes

∴ whatever data you give, it will ~~count~~ show you 'Yes'.

## ⊗ OneR Classifier:

consider any one feature ~~to~~ for prediction.




⇒ This will be the final model for prediction.

## Lecture - 7

### Classification Evaluation - 1

For ZeroR

$$\text{accuracy} = \frac{\overbrace{1+1+0+1+1}^{\text{Yes count}}}{\underbrace{5}_{\text{Total Sample}}} = \frac{4}{5} = 80\%$$

Let's

Given 1000 apple, target to find good or bad

good = 950
bad = 50

Apply ZeroR ⇒ good

$$\text{accuracy} = \frac{950}{1000} = 95\%$$

But, this model can't detect the bad apple.

## ⊛ Confusion Matrix:

Actual value

| Predicted Value | | P 00 | 01 N |
|---|---|---|---|
| | P | TP<br>True Positive | FP<br>False Positive |
| | N | FN<br>False Negative | TN<br>True Negative |
| | | 10 | 11 |

~~0 → No~~

~~1 → Yes~~

⇒ for 2 target matrix 2×2

(i) Precision $= \dfrac{TP}{TP + FP}$

if $FP = 0$
then Precision is 100%.

$FP > 0$ ; Precision < 100%.

$FP \uparrow \Rightarrow$ Precision $\downarrow$

$FP \downarrow \Rightarrow$ Precision $\uparrow$

(ii) recall $= \dfrac{TP}{TP + FN}$

$FN \uparrow \Rightarrow$ recall $\downarrow$

⊛ Precision ⇒ FP

recall ⇒ FN

~~Precision ↑ ⇒ Recall ↓~~

# Lecture- 8

## Classification Evaluation -2

Precision ⇒ FP

Recall ⇒ FN

Like, for medical False Negative is very dangerous. User will not get treatment timely lead to high complexity in his/her body.

So, we need to work with recall.

For spam detection we can ~~w~~ work with precision.

⊛ In general, if we can focus in both then we will get a better model. For that we use F1 score.

$$F_1 \text{ Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{recall}}$$ ⇒ P.R. harmonic mean

close to small value

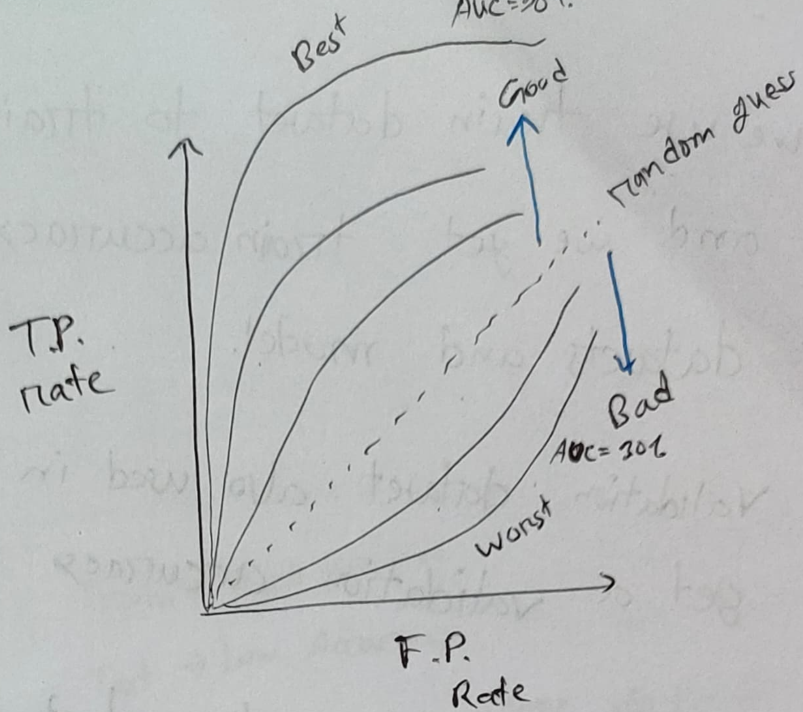
two number 40, 100

$$\text{mean} = \frac{40+100}{2} = 70$$

$$\text{H. mean} = \frac{2 \times 40 \times 100}{40+100}$$

$$= 57.14$$

## ⊛ ROC - Curve:




AUC ⇒ Area under the Curve

AUC ↑ ⇒ Better model.

⊛ Accuracy = $\dfrac{TP + TN}{TP + TN + FP + FN}$

$TP + TN$ → all correct, sum

$TP + TN + FP + FN$ → sum of all

## Lecture - 9

Bias ~~and und~~ Variance

overfitting and underfitting

First we need to divide the dataset into three sets.

i. Train
ii. validation

} most of the cases combined in one datasets 80%.

iii. Test 20% → Standard

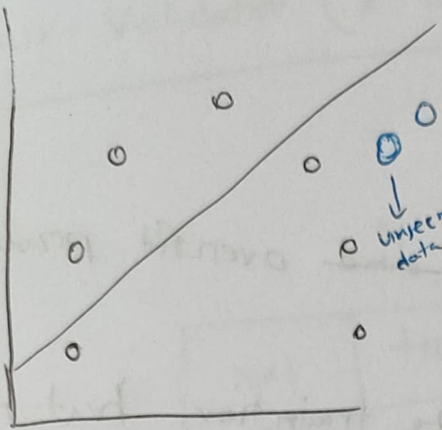
Paloxi
Palonosetron USP

⇒ we use train dataset to train our model. and we get train accuracy from the datasets and model.

Validation dataset also used in train and we get a validation accuracy from this dataset.

⇒ we need to keep the test datasets totally isolated from the model. so that the datasets works as an unseen datasets for the model. In reallife our model need to face a lots of unseen data. That's why after training we need to test our model by the test datasets.

And we need to find the test accuracy.



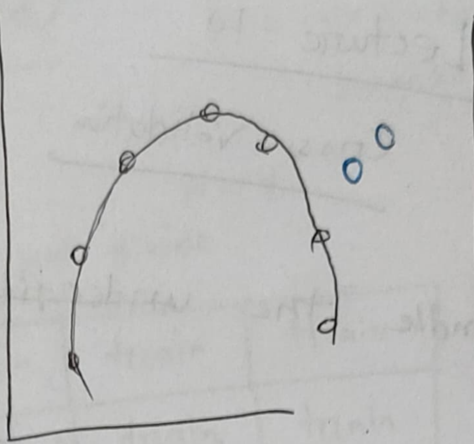

Train ⇒ Low Accuracy

Test ⇒ Low Accuracy

⇒ No accuracy

⇒ High Bias

⇒ Underfitting



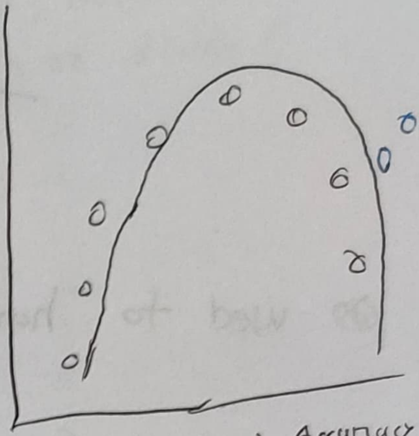
Train ⇒ High Accuracy

Test ⇒ Low Accuracy

⇒ Not ~~constent~~ consistency

⇒ High Variance

⇒ Overfitting



Train ⇒ Good Accuracy

Test ⇒ Good Accuracy

⇒ Perfect fit

- Low Bias
- Low Variance

Some Example:

Bias ⇒ opposite of accuracy

Variance ⇒ opposite of consistency

| | | | | | |
|---|---|---|---|---|---|
| Train = | 90 | 80 | 30 | 90 | 85 |
| Test = | 50 | 76 | 28 | 82 | 83 |
| | Overfit | good | Underfit | | Perfect |

good

overfit

need to retrain
or test with another unseen data.

# Lecture - 10

## Cross - Validation

- used to handle the ~~underfit and~~ overfit problem.

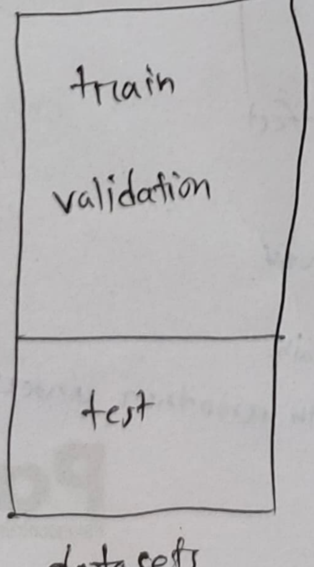
For underfit ⇒ we can increase the train time but not guaranteed that it will solve the problem.

- can increase the feature by feature engineering.

For Overfit ⇒

we can use cross validation

- Regularization
  - lasso
  - Ridge
- elastic net




datasets

if the test data pattern is totally different than train then the overfitting issue occurs.

## Cross - Validation (k-fold):

k = variable, hold the number of divide

$k = 4$

4 equal divide

| | | | |
|---|---|---|---|
| val | train | train | train |
| train | val | train | train |
| train | train | val | train |
| train | train | train | val |
| 70% | 90% | 30% | 70% |

$$\text{average} = \frac{70 + 90 + 30 + 70}{4} = 70\%$$

= validation accuracy

⇒ Now this model is ready to test.