7 - steps

1) Form a Hypothesis

↳ We can predict how many medals a country will win in the olympics.

2) Find the DATA

Team	Year	Athletes	Prev Medals	Medals
USA	2008	763	263	317
USA	2012	689	317	248
USA	2016	719	248	264
IND	2008	67	1	3
IND	2012	95	3	6
IND	2016	130	6	2

3. Reshape the DATA

⇒ Once we have data need to reshape it to make it ML predictions possible.

Team	Year	Athletes	Prev Medals	Medals
USA	2008	763	263	317
USA	2012	689	317	248
USA	2016	719	248	264
IND	2008	67	1	3
IND	2012	95	3	6
IND	2016	130	6	2

⇒ In this case everything is available in single row, we can use Athlete & Prev medal column to predict how many medals they can win in the given year.

(4) Clean the DATA

Team	Year	Athletes	Prev Medals	Medals
ALB	1992	9	-	0
ALG	1964	7	–	0
AND	1976	3	-	0
BLR	1996	259	-	23
ARM	1996	38	-	2

⇒ Most ML models cannot work with missing values hence need to clean

(5) Error Metric

$$\sum_{i=1}^{D} |x_i - y_i|$$

Team	Year	Medals	Predictions	Error
ALB	1992	0	3	3
ALG	1964	0	2	2
AND	1976	0	2	2
BLR	1996	23	15	8
ARM	1996	2	5	3

⑥ Split the DATA

Training Data

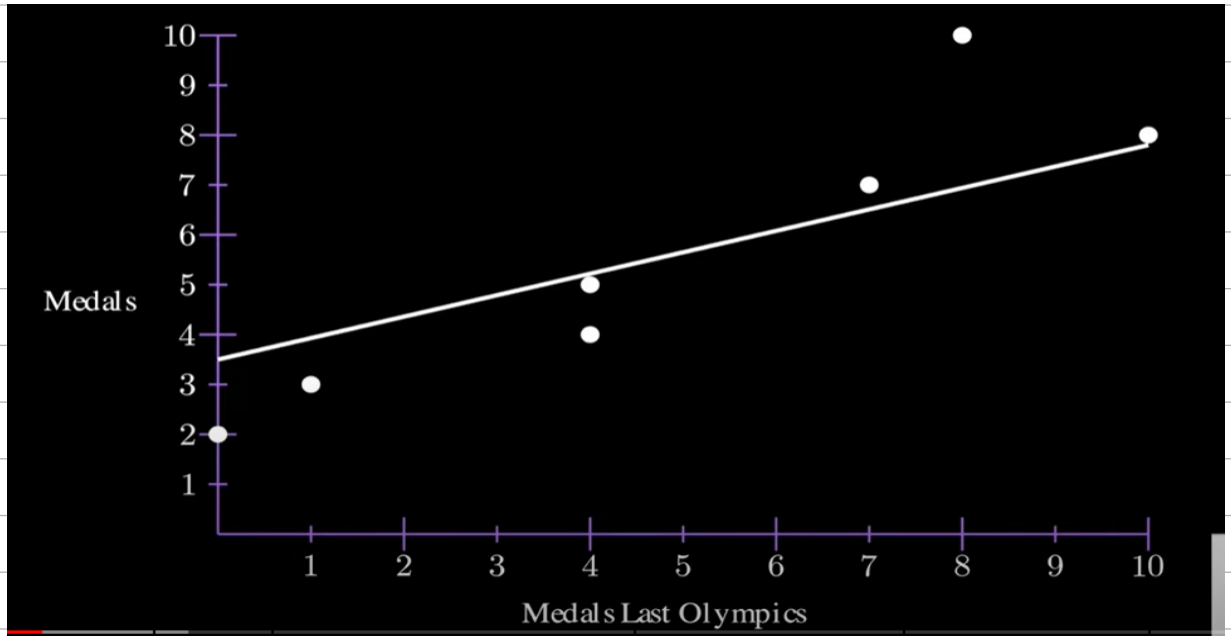
Team	Year	Athletes	Prev Medals	Medals
USA	2008	763	263	317
USA	2012	689	317	248
IND	2008	67	1	3
IND	2012	95	3	6

Test Data

Team	Year	Athletes	Prev Medals	Medals
USA	2016	719	248	264
IND	2016	130	6	2

⑦ Train a model

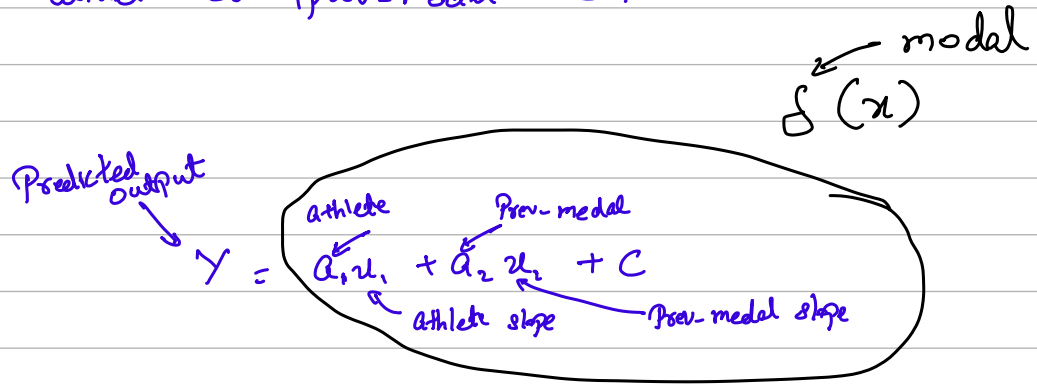
$$y = ax + B$$



$$y = a_1x_1 + a_2x_2 + B$$

Approch :

⇒ we are trying to predict medal on the basis of athlete or prev_medal so.



⇒ now need to check is there any correlation in between a_1 & Y || a_2 & Y

⇒ teams.corr() ["medals"]

↳ it will take medal as 1 & relate other columns

Years	−0.02...
athletes	0.84...
age	0.02 ...
prev_medal	0.92 ...
medals	1.000

" you can observe athles & prev medal is near "

⇒ always check that parameter what you are going to predict how balance is it. here

teams.plot.hist (Y = "medals")

⇒ now remove all the null values

⇒ Split the data for train & test

train data set < 2012

test data set >= 2012

⇒ Built model

Initiate — reg object initiated

Build — reg.fit(predictors, target)

predict — predict with test call with reg.predict function

what we observe our predicted values are in decimals & negatives.

```
test.loc[test["predictions"]<0, "predictions"] = 0
```

← for removing negative numbers

```
test["predictions"] = test["predictions"].round()
```

↳ for making it round

⇒ Now actual medal & our medal predictions readily need to check Error (mean_absolute_error)

⇒ So, i got 3.29 MAE, it means on average our predictions is 3.29.

⇒ always check your error is lower than your std deviation