(X) → = Independent / predictor / Explanatory

Y = Dependent / response variable. Date / /

Linear Regression (for prediction or forecasting)

- Dependence of one variable on one or more variables (called independent variables).

⇒ Primary used to input to a system

- Dependent & independent variables kay relationship ka estimate.
- Effect of each explanatory variables on dependent variables.
- Predict value of dependent variable for given ind variable.

⇒
- Dependent are those which are changed due to indep.
- least square linear regression ⇒ method for determining Y on basis of X.

⇒ **Assumptions of Linear Regression Model.**

- Linear Functional form
- Fixed independent Variables
- Equality of varience of the errors.
- No multicollinearity
- No outlier distortion
- No autocorrelation of the errors.

⇒ **Linear Regression Model.**

First order linear model

$$Y = b_0 + b_1 X + \varepsilon$$

b_0 = Y-intercept

b_1 = slope of the line

ε = error variable.

slope. Y-intercept.

$y = mx + b$

predict
↑
- Independent change, 1 dependent change
- 2 Independent change, 1 dependent change

Date / /

- If there is only one driver variable, X, it is simple linear regression.
- Model involves multiple driver variables called multiple linear regression.

- If Relationship b/w X and Y is curvilinear, the regression line will be be a curved line

- Greater strength of relationship b/w X and Y better is prediction.

⇒ **Least Squares Estimation of b_0, b_1**

Least Square estimates of slope co-efficient b_1 of true regression line

② $SSE = \sum_{i=1}^{n} \left(y_i - \left(\hat{\beta}_0 + \hat{\beta}_1 x_i\right)\right)^2$

① $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$

② $\beta_1 = \dfrac{\sum \left(x_i - X'\right)\left(y_i - Y'\right)}{\sum \left(x_i - X'\right)^2}$

β_0 = Mean response when $x = 0$

β_1 = change in mean response when x increases by 1 unit.

$$\beta_0 = Y' - \beta_1 X'$$

- β_0, β_1 are unknown parameters.

⇒
- Regression generates least squares regression line.
- Squaring difference and adding up squared differences across all predictions is called residual / error sums / squares / SSerror.
- Regression generates formula such that SSerror is as small as it can possibly be.
- Minimizing this number minimizes average error.

Date / /

Regression in Python

1. from sklearn.linear_model import LinearRegression
   Do transformation. Use reshape(), set x, y
2.
3. • model = LinearRegression().fit(x,y)

• fit_intercept ⇒ Boolean, if true decides to calculate intercept bo.
(By default true) • if false consider it equal to zero.

• normalize ⇒ Boolean, if true decides to normalize the input variables
(By default false)
↓
It doesn't normalize input variables.

model.intercept_ (bo)
model.coef_ (b1)

4. Check results of model fitting
• Obtain co-efficient of determination, R^2 with .score().
r1 = model.score(x,y)
pre = model.predict(x)

Applications

Economic growth
Product sale
Housing Sales
Score prediction.

If $K == n$ (no. of obs), then distance = 0 (case of overfitting)

K-mean Clustering

Date / /

	x	y	
1	1.0	1.0	→ Mean 1
2	1.5	2.0	
3	3.0	4.0	
4	5.0	7.0	→ Mean 2
5	3.5	5.0	
6	4.5	5.0	
7	3.5	4.5	

Advantages:-
- Easy to represent
- Can work in multiple dimension

Disadvantages:-
- Time consuming to find optimal number of clusters.
- Time consuming feature engineering

Iteration 1:-

=> For point 1:- $x_1 = 1.0$, $y_1 = 1.0$

$$D_1 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
$$= 0$$
$$D_2 = \sqrt{(5-1)^2 + (7-1)^2}$$
$$= \sqrt{(4)^2 + (6)^2}$$
$$= \sqrt{16+36}$$
$$= \sqrt{52} = 6.10$$

=> For Point 2:- $x_1 = 1.5$, $y_1 = 2.0$

$$D_1 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
$$D_1 = \sqrt{(1-1.5)^2 + (-2)^2}$$
$$= \sqrt{(-0.5)^2 + (1)^2}$$
$$= 1.12$$
$$D_2 = \sqrt{(5-1.5)^2 + (7-2)^2}$$
$$= 6.10.$$

Point	D_1	D_2	Which cluster to choose.
1 (1.0, 1.0)	0	6.10	1
2 (1.5, 2.0)	1.12	6.10	1