Statistics - Science of collecting, organizing, analyzing and interpreting data

Data
- Nominal
- Ordinal
- Interval

Variable
- Quantitative
  - Discrete → no. of students
  - Continuous → Weight, Height
- Qualitative / Categorical
  - Ordinal → Feedback 1-10, Ranks
  - Nominal → Eye color, Blood Group

* Machine learning

=> Supervised learning
=> Un supervised learning

=> Reinforcement learning : Feedback based.

General M.L. Model

I/P → Machine → Output

Feedback.

⇒ M.L. is possible because of large data available & Computation power.

⇒ Application :- Healthcare
:- Sentiment analysis
:- Fraud detection

⇒ M.L. is the science of making computer learn & act like humans by feeding data & information without being explicitly coded / programmed.

★ Steps to M.L. project [general]

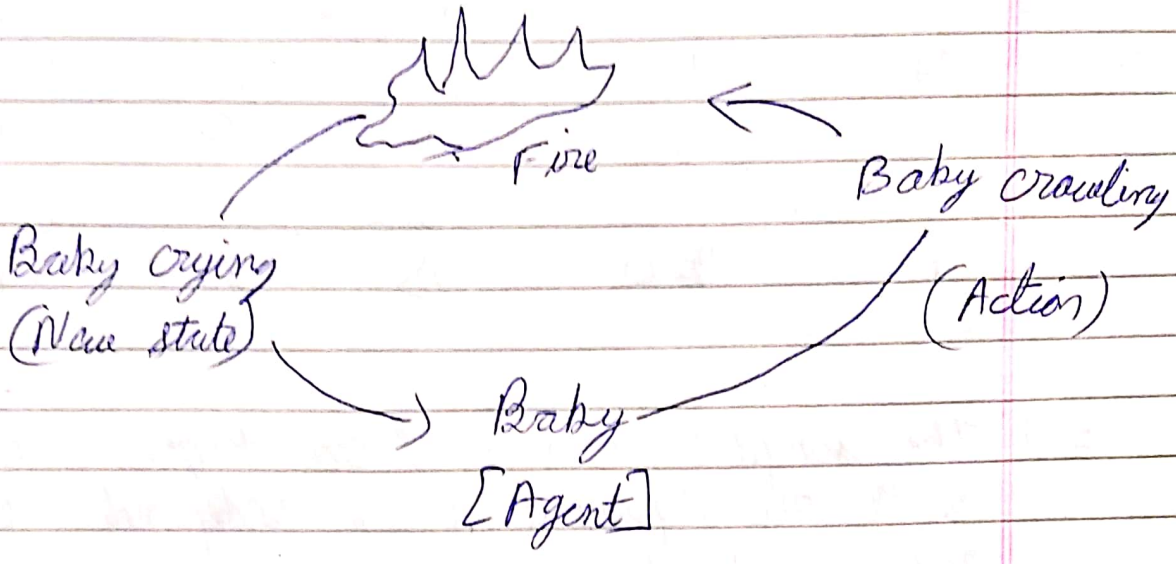
→ Define your objective
→ Collecting data
→ Prepare data
→ Select Algorithm
→ Train Model
→ Test Model
→ Predict
→ Deploy

★ Division based on what M.L. does.

1) Want to predict category :- Classification
2) Want to predict quantity :- Regression.
3) Detect anomaly :- anomaly detection

Date
Page

★ Reinforcement learning is an important type of M.L. where an agent learn how to behave in an environment by performing action & see the results.



★ Supervised & Unsupervised

- Labled data	→ Non-labled data
- Direct feedback	→ No feedback
- Predict Output	→ Find hidden structure in data

★ Linear Regression

=) A statistical model used to predict the relationship between independent & dependent variables

Examine two factors

=) Which variables in particular are significant predictors of the outcome variables

=) How significant is the Regression line to make predictions with highest possible accuracy.

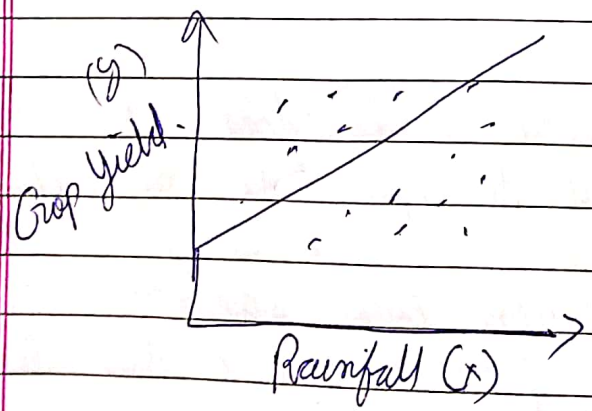
Linear Regression

x	y	(x^2)	(y^2)	$(x*y)$
1	2	1	4	2
2	4	4	16	8
3	5	9	25	15
4	4	16	26	16
5	5	25	25	25
Σ	15	40	55	86	66

⇒ The simplest form of linear Regression equation with one dependent & one independent variable is represented by:-

$y = m * x + c$

y --> Dependent variables
x --> Independent variables
m --> Slope of the line
C --> Coefficient of line



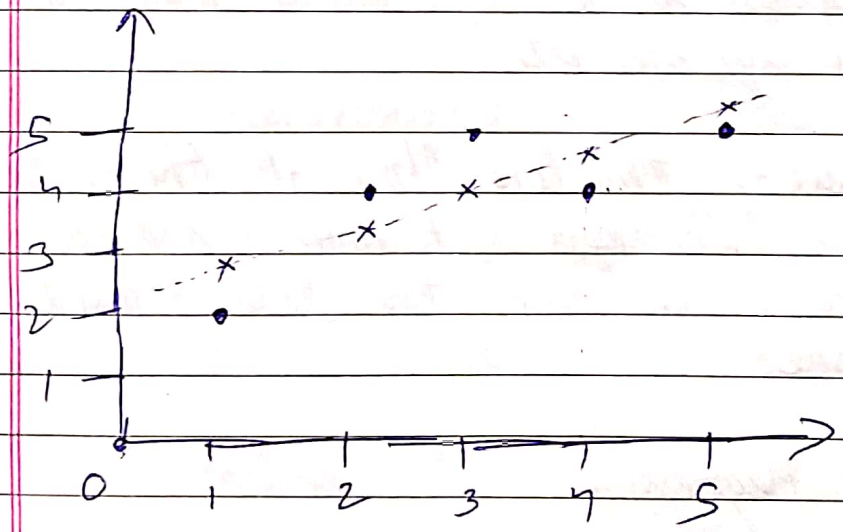
$$m = \frac{((n * \Sigma(x*y)) - (\Sigma(x) * \Sigma(y))}{((n * \Sigma(x^2)) - (\Sigma(x)^2)}$$

$$= \frac{((5*66) - (15*26))}{((5*55)) - (225)} = 0.6$$

$$c = \frac{((\Sigma(y) * \Sigma(x^2)) - (\Sigma(x) * \Sigma(x*y))}{((n * \Sigma(x^2)) - (\Sigma(x)^2)}$$

$$= 2.2$$

$$y = m * x + c$$
$$= 0.6 * 3 + 2.2$$
$$= 4$$



y_{pred}

$y = 0.6 * 1 + 2.2 = 2.8$
$y = 0.6 * 2 + 2.2 = 3.4$
$y = 0.6 * 3 + 2.2 = 4$
$y = 0.6 * 4 + 2.2 = 4.6$
$y = 0.6 * 5 + 2.2 = 5.2$

L.R. :- Works for continuous value
Linear : Supervised learning

X	Y	Y_{pred}	$(Y - Y_{pred})$	$(Y - Y_{pred})^2$
1	2	2.8	-0.8	0.64
2	4	3.4	0.6	0.36
3	5	4	1	1
4	4	4.6	-0.6	0.36
5	5	5.2	-0.2	0.04
				$\Sigma = 2.4$

⇒ The sum of squared errors for this regression line is 2.4. We check this error for each line and conclude the best fit line having the least e square value.

⇒ Minimizing this distance, there are lots of ways, between line and data points like sum of squarred errors, sum of absolute erro Root mean square etc.

⇒ We keep moving this line through the data poin to make sure the best fit lines have the leas square distance between the data points & regression line.

Multi linear Regression

L.R. $y = m * x + C$

M.L.R. $y = m_1 x_1 + m_2 x_2 + m_3 x_3 + \ldots + C$

Slopes

Dependent Variable

Independent Variable

Co-efficient

Implementation of L.R

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
%. matplotlib inline
```

```
# importing the data set & extract dependent and independent variable

companies = pd.read_csv('          .csv')

X = companies.iloc[:, :-1].values
y = companies.iloc[:, 4].values

companies.head()
```

```
# data virtualization, building correlation matrix

sns.heatmap(companies.corr())
```

```
# encoding categorical data

from sklearn.preprocessing import labelencoder, OneHotEncoder
labelencoder = LabelEncoder()

X[:, 3] = labelencoder.fit_transform(X[:, 3])
```

onehotencoder = OneHotEncoder(categorical_features = [3])

x = onehotencoder.fit_transform(x).toarray()

#print(x[0])

x = x[:, 1:]

- # Spliting of our data
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 0)

- from sklearn.linear_model import LinearRegression

regressor = LinearRegression()

regressor.fit(x_train, y_train)

- # Predicting the test set results

y_pred = regressor.predict(x_test)

y_pred =

- # Calculating co-efficient & intercept

  print (regressor.coef_) # [gives slope]

  print (regressor.intercept) # [gives constant]

- # Calculating R squared matrix (evaluting our model)

  from sklearn.metrices import r2_score
  r2_score (y_test, y_pred)

Logistic Regression

⇒ An algorithm to perform an binary classification

⇒ It is of type which gives discrete values

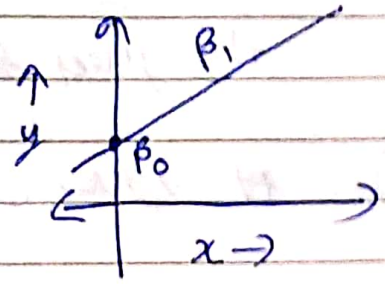
⇒ Supervised learning

⇒ Odds of success(Θ) = $\frac{\text{Probability of an event happening}}{\text{Probability of an event not happening}}$

Or $\Theta = \frac{P}{1-P}$

range of values of odds is 0 to ∞
values of probability change from 0 to 1

* Maths behind logistic Regression



Here β_0 is the y-intercept, β_1 is slope of the line x is value of the x-coordinate y is value of prediction

eg:- $y = \beta_0 + \beta_1 x$

Now, we predict the odds of success

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$$

exponenting both sides

$$e^{\ln\left(\frac{p(x)}{1-p(x)}\right)} = e^{\beta_0 + \beta_1 x}$$

$$\frac{p(x)}{1-p(x)} = e^{\beta_0 + \beta_1 x}$$

Let $y = e^{\beta_0 + \beta_1 x}$

then $\frac{p(x)}{1-p(x)} = y$

$$p(x) = y(1-p(x))$$

$$= y - y\,p(x)$$

$$p(x) + y(x) = y$$

$$p(x)(1+y) = y$$

$$p(x) = \frac{y}{1+y}$$

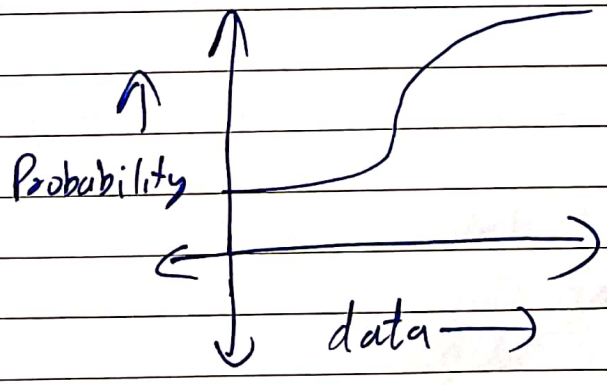
$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

The equation of a sigmoid function

$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

$$\boxed{p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}}$$

equation of logistic Regression that is sigmoid function



⇐ Example of sigmoid curve

Linear Regression	Logistic Regression
⇒ Used to solve Regression problems	⇒ Used to solve classification problem
⇒ The response variable are continuous in nature	⇒ The response variable is categorical in nature
⇒ It helps to ~~calculate~~ estimate dependent variable where their is a change in Independent variable	⇒ It helps to calculate the possibility of a particular event taking place
⇒ It is a straight line	⇒ An S-curve (S= Sigmoid)
⇒	⇒

Use-Case implementation of logistic Regression

```
# Importing lib.
from sklearn.datasets import load_digits
from sklearn.model_selection import
                                train_test_split
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn import metrics
%. matplotlib inline
digits = load_digits()

# determining the total number of images & labels

print("Image Data shape", digits.data.shape)
print("Label Data shape", digits.target.shape)

# Displaying some of the images & labels

plt.figure(figsize=(20,4))

for index, (image, label) in enumerate
        (zip(digits.data[0:5],
             digits.target[0:5])):

    plt.subplot(1, 5, index+1)
    plt.imshow(np.reshape(image, (8,8)),
               cmap=plt.cm.gray)
    plt.title('Training: %i \n' % label,
              fontsize = 20)
```

dividing dataset into training & test set

```
x_train, x_test, y_train, y_test =
        train_test_split(digits.data, digits.target,
                         test_size=0.23,
                         random_state=2)
```

import the logistic Regression model

```
from sklearn.linear_model import LogisticRegression
```

Making an instance of the model &
training it

```
logisticRegr = LogisticRegression()
logisticRegr.fit(x_train, y_train())
```

Predicting output of 1st element of test set

```
print(logisticRegr.predict(x_test[0].
                           reshape(1,-1)))
```

Predicting the output of first 10 element of test

```
logisticRegr.predict(x_test[0:10])
```

Prediction for the entire dataset

```
predictions = logisticRegr.predict(x_test)
```

determining the accuracy of the model

```
score = logisticRegr.score(x_test, y_test)
print(score)
cm = metrics.confusion_matrix(y_test, prediction)
# confusion matrix

plt.figure(figsize=(9,9))
sns.heatmap(cm, annot=True, fmt=".3f",
            linewidth=.5, square=True,
            cmap='Blues_r');
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
all_sample_title = 'Accuracy score: {0}'.
                   format(score)
plt.title(all_sample_title, size=15);
```

Date
Page

★ ANOVA :- Analysis of Variance by R.A. Fisher

⇒ A statistical test for detecting difference in the group means when there is one parametric dependent variable & one or more independent variable

⇒ It is used when we want to study 3 or more population study.

Example :- NY	Texas	Oregeon
18	18	21
19	20	22
20	16	17
21	20	18
22	21	22
23	20	19
18	18	21
19	19	20
20	17	18
21	13	23
$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$

H_0 :- $\mu_1 = \mu_2 = \mu$

H_a : at least one mean is different

Grand mean :- The mean of the sample means

$$\bar{\bar{x}} = \frac{\sum_{i=1}^{k} \left[\sum_{j=1}^{n_i} x_{ij} \right]}{\sum n_i}$$

K = population set {3}

n_i = sample size of i^{th} - population

$$\bar{\bar{x}} = \frac{584}{30} = 19.466667$$

Null hypothesis : A general statement that stater that there is no relationship between two measured phenomena or no association among groups

Alternate hypothesis :- It states whenever something is happening, a new theory is prefered instead of an old one.

P value :- Probability of finding the observed, or more extreme, results when the null hypothesis of a study question is true. {P}

Alpha value :- Criterion for determining whether a test statistic is statistically significant {α}

ΣN	x^2	mN	x^2	ΣN	x^2
10	100	8	64	4	16
9	81	4	16	3	9
6	36	6	36	6	36
7	49	7	49	4	16
Σx_1 = 32	$266 = \Sigma x_1^2$	Σx_2 = 25	Σx_2^2 = 165	Σx_3 = 17	$\Sigma x_3^2 = 77$

Correction Term : $C_x \frac{(\Sigma x)^2}{N} = \frac{(32+25+17)^2}{12}$

$= 456.33$

Sum of Square of total : $SS_T = \Sigma x^2 - C_x =$

$= (266 + 165 + 77) - 465.33$

$= 51.67$

Sum of Square among groups

$$SS_A = \frac{(\Sigma X)^2}{n} - C_x = \left(\frac{32^2}{4} + \frac{25^2}{4} + \frac{17^2}{4}\right) - 465.33$$

$$= 28.17$$

Sum of Squares within groups

$$SS_w = SS_T - SS_A$$
$$= 51.67 - 28.17 = 23.5$$

Mean of Sum of Squares among groups

$$MSS_A = \frac{SS_A}{k-1} = \frac{28.17}{3-1} = 14.085$$

degree of freedom among group → $k-1$

Mean of Sum of Squares within groups

$$MSS_w = \frac{SS_w}{N-k} = \frac{23.5}{12-3} = 2.611$$

degree of freedom within group → $N-k$

N ... Total no of samples
k ... total categories or group

$$F \text{ Ratio} = \frac{MSS_A}{MSS_w} = \frac{14.085}{2.611} = 5.394$$

$F(2, 9)$ 5.394 $\quad \alpha = 0.05$ ①
$F(2, 9)$ 5.34 $\quad \alpha = 0.01$

Standard deviation σ

=) Our varies from average. Low SD. means maximum data is near average & high S.D. means its not so

Population

$$\sigma = \sqrt{\frac{\Sigma (x-\bar{x})^2}{n}}$$

Sample

$$\sqrt{\frac{\Sigma (x-\bar{x})^2}{n-1}}$$

↑ Bessel's Correction

$$\text{variance} = \frac{\Sigma (x-\bar{x})^2}{n}$$

① Reject null hypothesis & accept Alternate hypothesis

95-1. confidence

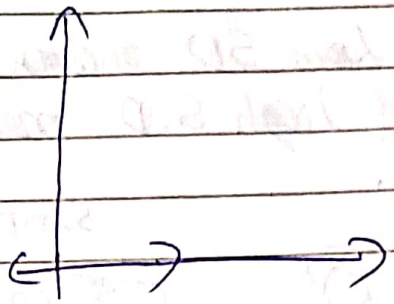
② Reject alternate hypothesis & accept null hypothesis

99-1. confidence

H_0 : No significant effect of noise on number of question solved.

H_1 : Significant effect of noise on number of question solved.

Cost Function [Continuation of Linear regression]

⇒ 

Idea is to choose θ_0, θ_1 so that $h_\theta(x)$ is close to y for our training example (x, y)

$m \rightarrow$ # training example

⇒ minimize (θ_0, θ_1) $\frac{1}{2m} \sum_{i=0}^{m} (h_\theta(x) - y)^2$

$J(\theta_0, \theta_1)$

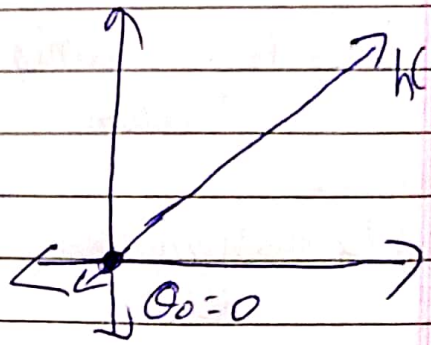
Cost function also squared error function

⇒ hypothesis $h_\theta(x) = \theta_0 + \theta_1(x)$
Parameter θ_0, θ_1

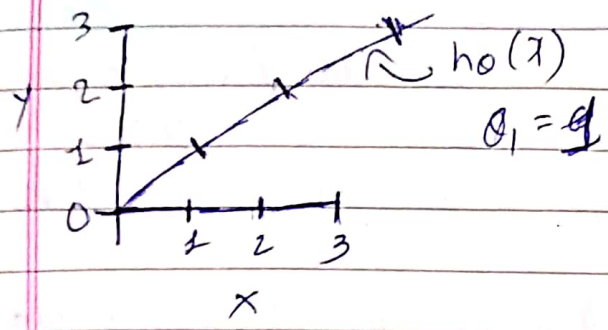
Goal : minimize $J(\theta_0, \theta_1)$

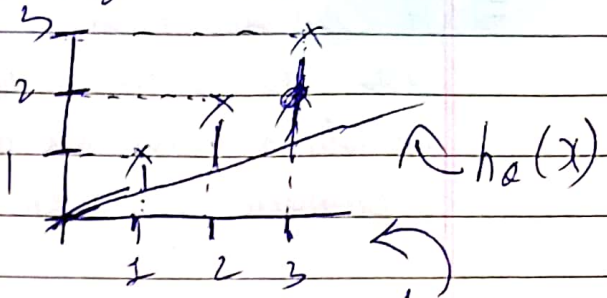
⇒ Example let consider $h_\theta(x) = \theta_1 x$, $\theta_0 = 0$

That is



$h_\theta(x)$ for fixed θ_1, this is a function of x





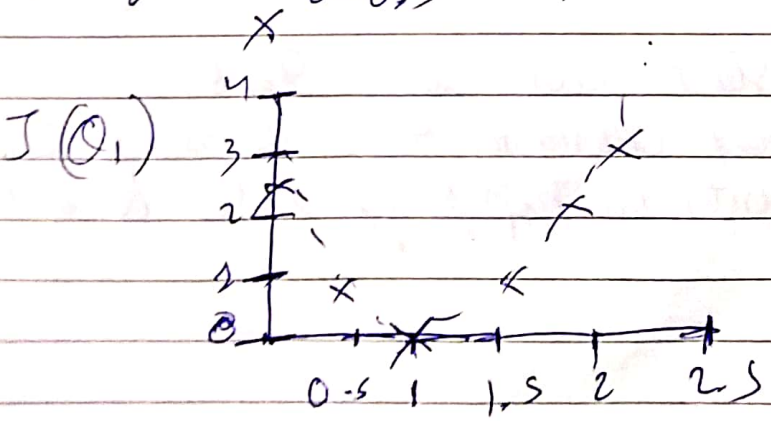
for $\theta_1 = 0$ then $h(x)$ = x axis

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$= \frac{1}{2m} \sum_{i=1}^{n} \left(\theta_1 x^{(i)} - y^{(i)}\right)^2$$

$$= \frac{1}{2m} \left(0^2 + 0^2 + 0^2\right) = 0^2$$

$J(\theta_1)$ function of the parameter θ_1



$\theta = 0.5$

$$J(0) = \frac{1}{2m}\left(1^2 + 2^2 + 3^2\right)$$

$$= 2.3$$

$$J(0.5) = \frac{1}{2m}\left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right]$$

$$= \frac{1}{2\times 3}(3.5) = 0.58$$

⇒ A contour plot is a graph that contains many contour lines. A contour lines of a two variable function has a constant value at all points of the same line. An example of such a graph is the one to right below

⇒ If we have one parameter, hence can be described using 2D graph that is parameter vs $J(\theta)$ the cost function

⇒ For more parameter dimension shall increase and hence difficult to visualize

★ Gradient descent

⇒ As a general function, for minimizing cost function in linear regression, also to minimize other function

⇒ We have $J(\theta_0, \theta_1)$ we want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline ⇒ start with some θ_0, θ_1

⇒ Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum

Algorithm

repeat until convergence

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad \text{for } j = 0 \& j = 1$$

Correct Simultaneous update

$temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$\theta_0 = temp0$

$\theta_1 = temp1$

α ---- learning rate

Multivariate Linear Regression

⇒ Previously :- $h_\theta(x) = \theta_0 + \theta_1 x$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

n = no. of feature

$x^{(i)}$ = input features of i^{th} training example

$x_j^{(i)}$ = value of feature j in i^{th} training example.

Clustering

⇒ Grouping data points & creating partitions based on similarities

⇒ Donot have response class

Types :- Centroid Based clustering
1) K-means

:- Connectivity Based Clustering

:- Distribution Based Clustering
:- Density based Clustering

⇒ Sensitive to outliers
⇒ We need to supply the number of Cluster

Examples :- Applications of Clustering

:- Market Segmentation
:- Social Network analysis
:- Astronomical data analysis.

K- Means

⇒ Input :- No. of cluster that is value of K
:- Training set

$K = \{2, 3, 4, 10, 11, 12, 20, 25, 30\}$ for $K=2$

$m_1 = 4$ $\quad$ $m_2 = 12$

$K_1 = \{2, 3, 4\}$ $\quad$ $K_2 = \{10, 11, 12, 20, 25, 30\}$

Now $m_1 = 3$ $\quad$ $m_2 = 18 = \frac{108}{6}$

$K_1 = \{2, 3, 4, 10\}$ $\quad$ $K_2 = \{11, 12, 20, 25, 30\}$

$m_1 = 4.75$ $\quad$ $m_2 = 19.6$

$K_1 = \{2, 3, 4, 10, 11, 12\}$ $\quad$ $K_2 = \{20, 25, 30\}$

$m_1 = 7$ $\quad$ $m_2 = 25$

$K_1 = \{2, 3, 4, 10, 11, 12\}$ $\quad$ $K_2 = \{20, 25, 30\}$

$m_1 = 7$ $\quad$ $m_2 = 25$

⇒ Thus we are getting same means.

K- Mediods

i	x	y
x_1	2	6
x_2	3	4
x_3	3	8
x_4	4	7
x_5	6	2
x_6	6	4
x_7	7	3
x_8	7	4
x_9	8	5
x_{10}	7	6

① C_1 (3, 4) ② C_2 (7, 4)

$|a-c| + |b-d|$

① $|2-3| + |6-4| \Rightarrow 3$

$0 + 4 \Rightarrow 4$

$1 + 3 \Rightarrow 4$

$3 + 2 \Rightarrow 5$

$3 + 0 \Rightarrow 3$

$4 + 1 \Rightarrow 5$

$5 + 1 \Rightarrow 6$

$4 + 2 \Rightarrow 6$

② $|2-7| + |6-4| \Rightarrow 7$

$4 + 4 \Rightarrow 8$

$3 + 3 \Rightarrow 6$

$1 + 2 \Rightarrow 3$

$1 + 0 \Rightarrow 1$

$0 + 1 \Rightarrow 1$

$1 + 1 \Rightarrow 2$

$0 + 2 \Rightarrow 2$

Step II then cluster are

C1: {(2,6), (3,8), (4,7), (3,4)}
C2: {(7,4), (6,2), (6,4), (7,3), (8,5), (7,6)}

$$\text{Total Cost} = \sum_{(x,c)}^{d} \sum_{i=1} |x_i - c_i|$$

$$= \{ \text{Cost}(3,4),(2,6) \quad \text{Cost}(7,4)(8,5)$$
$$\text{Cost}(3,4)(3,8) \quad \text{Cost}(7,4)(7,6)$$
$$\text{Cost}(3,4)(4,7) \quad \vdots$$

$$= (3+4+4) + (3+1+1+2+2)$$
$$= 20$$

Select one of Non-medoids O'

Let = (7,3) i.e. x_7

II) So new medoids are C(3,4) & O'(7,3)

	x	y	dist (7,3)	dist (3,4)	
x_1	2	6	8	(3)	
x_3	3	8	9	(4)	Total Cost
x_4	4	7	7	(4)	$\Rightarrow 3+4+4+2+2+1+3+3$
x_5	6	2	(2)	5	
x_6	6	4	(2)	3	
x_7	7	4	(1)	4	$\Rightarrow 22$
x_8	8	5	(3)	6	
x_{10}	7	6	(3)	6	