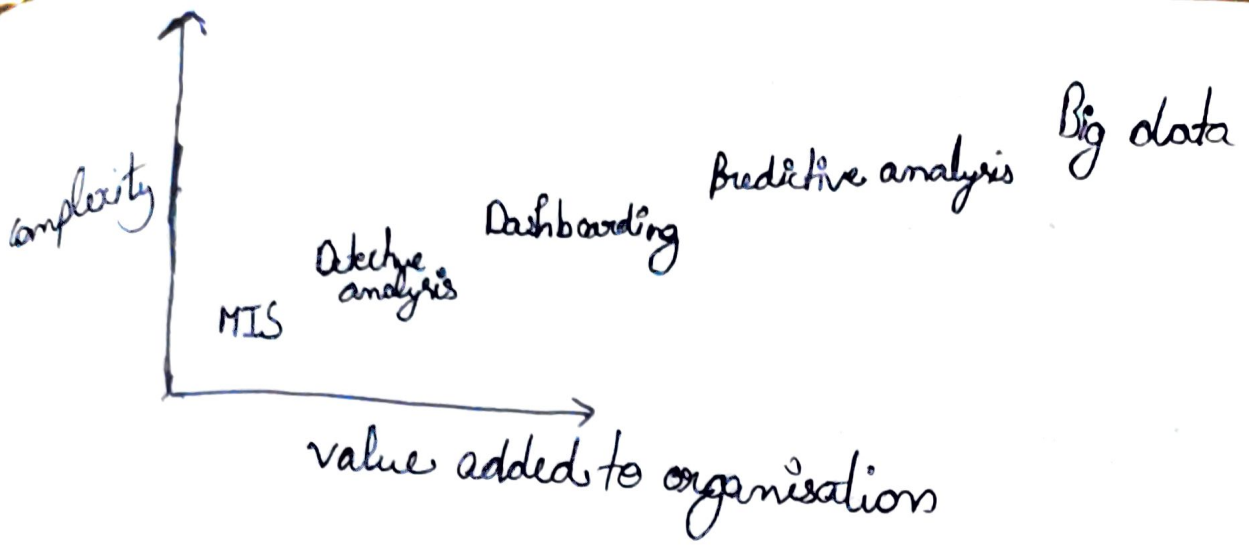


MIS (Management Info. sys) - (Data on) What has happened previously

Detective analysis - Why it has happened, understanding the scenario.

Dashboarding - Realtime analysis of what's happening to make business decisions.

Predictive modeling - What could happen in future based on past data

Big data - higher complicity of data unable to be handled by traditional system due to the volume, variety & velocity of data.

Forecasting : Predicting or estimating the future based on past and present data

eg: how many passengers are to be expected in a given flight.

Machine learning - Teach machine to learn things and improve predictions based on data on their own.

eg: • google search
• amazon recommendation.

Applications of data science

- recommendation system
- Social media - Follow/add recommendation,
  Ad placement
  sponsered feed.
- Banking - Credit Scoring
  fraud detection
  Price optimizn.
- e-commerce - Discount price optmzn.
  Cross sell up-sell
  Business forecasting
- Search engines - Search algorithm
  fraud detection
  Ad placement
  Personalized result.

Python

- Open source
- simple syntax
- large development community
- Availability of diverse packages

Operators

arithmetic → returns float

+, -, *, /, %,

// - returns integer

comparison

>, <, >=, <=, !=, ==

logical

and - Returns first encountered false, else last value

eg: 0 and 3 => 0
3 and 5 => 5

or - returns first encountered true value, else last value whichever.

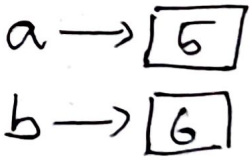
eg: 0 and 3 => 3
1 and 3 => 1
5 and 3 => 5

Not - returns inversed boolean value

- Datatypes

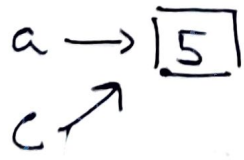
int, float, bool, str

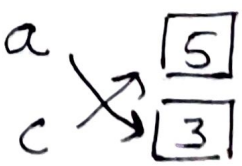
```
a = 5        // a points to memory location in which int value 5 is stored
b = a        // b points to same location as a
b = 6        // 6 is stored as int value in a different location and then b points to that location.
```



Now if

```
c = a
a = 3
```





Python does not have 'char' datatype; characters are treated as string

set(x) is used to convert string into set.

- conditional statements
  - if-else -> single condition
  - if-elif-else -> multiple
    - else if if (cond): statement

- looping
  - for loop - for i in range(1000):
    //creates a sequence 0 → 999 and executes for each
    for i in range(11, 50) → taken 50-1
    //num b/n 10 & 50
    for i in range(a 11, b 50, 2) → increment value
    // creates sequence from 11 to 50-1; incrementing by 2
    11 13 15 .... 47 49
    a b-1

- functions in python

built-in: print(), range(), min(), max().

user-defined: using 'def' keyword

eg: def func_name(arguments):
    statements
    ...
    return statement

- Data structures in python
  - problem with existing datastructure
    - data in variable stored in single format -> int or decimal or strings etc.
    - large memory overhead by creating large no of variables
    - unfit for storing large structured data

• Lists • Dictionary

Lists

- ordered data structure separated by comma & enclosed within square brackets

```
list1 = [2,3,4,5,6]
list2 = ["python", 'ruby', java']
list3 = [1, 'hey', 2, 3, 'Hi']
```

storage based on index positioning

```
list1[1] => 3
list1[1:4] => 3,4,5 // 'a' to 'b-1'
   a b
list1[-1] => 6
```

Functions:

append(): list1.append(4)

extend(): list1.extend([7,8]) => [2,3,4,5,6,7,8]
0 1 2 3 4 5 6

• appending list to a list

```
list1.append([7,8])
=> [2,3,4,5,6,[7,8]] => list1[5][1]
   0 1 2 3 4  5          // o/p is 8
              0 1
```

remove():

remove a specified value.

```
list1.remove(2)
// [3,4,5,6]
```

del():

```
del list1[3]
```

list1 => [2,3,4,5,6]
0 1 2 3

↳ o/p // [2,3,4,6]

- Dictionary

When data storage doesn't require any sequence and data is accessed using user defined key values

A dictionary is an unordered data structured with elements separated by comma and stored as key : value pair. A dictionary is stored within curly brackets.

```
dict1 = { 'Anandu' : 121, 'Ankit' : 163, 'CP' : 179 }
```

- Adding new element - key : value

```
dict1['Aeanu'] = 101   // This is added alphabetically in order
```

- update function... To add multiple elements into a dict

```
dict1.update({'Sunil' : 70, 'Ram' : 68})
```

adds item in alphabetical order ascending

```
dict1
    { 'Aeanu' : 101,
      'Anandu' : 121,
      'Ankit' : 163,
      'CP' : 179,
      'Ram' : 68,
      'Sunil' : 70 }
```

- deleting key : value element

```
=> del dict1['Aeanu']
```

Standard library, modules and packages in python

↓

collection of code that you get when installing python into system.
eg: built in functions, constants, datatypes, fileformats that can be read/write

By explicitely importing modules it is recognized its code is to be using in program instead of standard built-in libraries (if where in present).

Modules are design and implementation of specific function to be incorporated into a program.

In case of "from-import" form of import

- The syntax is : from <module-name> import <Identifier>
- The namespace of imported module becomes part of importing module.
- The identifier in module are accessed directly as: identifier.

If you have accidently used a name that is same as a built-in module, then it might be difficult to import that module, which is basically called name clash. This form of import does not prevent name clash.

Pandas

Pandas is powerful python data analysis toolkit for reading, filtering, manipulating, visualizing and exporting data

funtionalities:

- read different varieties of data.
- functions for filtering, seluting & manipulating data.
- Plotting data for visualization and exploration ~~tobase~~ purpose.

	Reader
CSV	read_csv ✓
JSON	read_json
HTML	read_html
Local clipboard	read_clipboard
MS Excel	read_excel ✓
HDF5 format	read_hdf
SAS	read_sas
SQL	read_sql
Google Big Query	read_gbq

sample:>>

```
impart pandas as pd
df = pd.read_csv("Test.csv")
dataframe storing variable

df.head()  // to display rows in dataframe
```

- Understanding dataframe

A dataframe is similar to excel workbook tabular datasheet

Basic functions:

- To know dimension of dataframe — Dataframe.shape
- Access top or bottom 'n' records — Dataframe.head(n) / .tail(n) (default = 5)
- Access all columns — Dataframe.columns
- Access data of one column — Dataframe["columnname"]
- Access data of multiple columns

Dataframe[["column1", "column2", ...]]

dataframe.shape returns the number of rows and columns in a dataframe respectively. So, the first element represents the number of rows and hence df.shape[0] returns number of rows in a dataset.

- Basic operations.

- Reading a spreadsheet and basic operations.

selecting rows by their positions

df.iloc[:5] (a to b-1; 0 to 4)

selecting columns by their position

df.iloc[:, :2] (rows : columns; select columns 0 to 2-1)

- mode value in a dataset, the value that is repeated most frequently

code:

```
mode_of_df = df 'Marks' .mode()
print(mode_of_df)
```

- Types of variables
  - Continuous : Take continuous numeric values
  - Categorical : has discrete values
    - Ordinal - When the discrete values have order or ranking b/n them eg: feedback
    - Nominal - do not have any order
      eg: gender

Practice : An analysis team is present which notes each footballer stays on the field and the number of goals they score (time)

goals - discrete
Time - continuous

- Mean value

```
mean_of_df = df['Marks'].mean()
```

it is not robust since on very large or small value can alter greatly the average.

outlier (value on diff scale than other values)

- Median
  - > given a series
  - > arrange in ascending order
  - > if no. of elements is odd, middle value is median
  - > if even median is calculated by taking average of $\frac{n}{2}$th and $\frac{n+1}{2}$th values.

```
>>: median_of_df = df['Overall Marks'].median()
```

- Calculating_quantiles

```
Q1 = df['Overall Marks'].quantile(0.25)
Q2 =          "          (0.5)
Q3 =          "          (0.75)
Q4 =          "          (1)
```

Spread

- Range - difference b/n largest & smallest value
  suffers from problems of outlier
- Interquartile range: Diff b/n 1st quartile & 3rd quartile (IQR)

```
max_in_df = df['Marks'].max()
min_in_df = df['Marks'].min()
// range
range = max_in_df - min_in_df
Q1 = df['Marks'].quantile(0.25)
Q3 = df['Marks'].quantile(0.75)
IQR = Q3 - Q1  // interquartile range
```

- Variance
  - It is the average squared deviations from the mean

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

>> varn = df['Marks'].var(ddof=0)

- Standard deviation

To compare the spread of data relative to the standard units.

$$std = \sqrt{\sigma^2} = \sigma$$

>> StndrdDevn = df['Marks'].std(ddof = 0)

Sample standard deviation (& population SD)

If the data is being considered a population on its own, we divide by the number of data points 'n' if the data is a sample from a larger population, we divide by one fewer than the number of data points in the sample, n-1

Frequency (table)

freq_df = df['Subject'].value_counts() # to find which subject frequently selected.

Histogram is for continuous variables whereas bar graph is for categorical variables.

Bin size in histogram should be same for all bins Each bin in it have same size. We can change number of bins but the size of each bin will be same.

matplotlib.pyplot - to plot graphs and visualize data in graph form

```
%matplotlib inline // to draw within jupyter notebook
```

to plot dataframe in a dataframe, say 'histogram'
we use the function

```
plt.hist(x = 'Overall marks', data = histogram)
```

plt → import as plt (identifier)
'Overall marks' → column name
histogram → dataframe to be visualized

No. of bins can be specified thus >>

```
plt.hist(x = 'Marks', data = histogram, bins = 5) //
```

5 bins on the x-axis

Properties of a histogram ~> for continuous variables

- The bins should be of equal size
- The bins should not overlap
- The height of the bar denotes the frequency
- we get an idea of the distribution from the histogram.
- Histogram are used for continuous data by putting bins to the numerical data.

Bar graphs - for categorical variables

Probability

Experiment : Whether a particular event occurs given a set of conditions.

Outcome : Result of a trial

Event : One or more outcome of trial

probability : likelyhood of an event

probability of getting two numbers on simultaneous roll of two dies such that their product is even is $\frac{18+9}{36} = \frac{3}{4}$

Representation of probabilities in form of graph is called probability mass function/distribution.

- Bernoulli trials : exactly

  Experiment has two outcomes

  probability distn. of number of successes in n bernoulli trials is known as Binomial distn.

  probability of outcome

  $$p(X=k) = {}^{n}C_{k}\, p^{k} q^{n-k} \qquad {}^{n}C_{k} = \frac{n!}{(n-k)!\,k!}$$

- Continuous random variables.

  eg: Amount of sugar in orange
  lifespan of a human.

  In this case, instead of checking the probability for each value we can calculate the probability of a certain thing being in a certain range.

  given figure is probability density function.

Central limit theorem

If we take means of random samples from a dist. and we plot the means, the graph approaches to a normal distribution when we have taken sufficiently large number of samples.

It also states that the mean of means will be approximately equal to the mean of sample means.

If the standard error of the sampling dist. is small, the sampling dist. of sample mean is approx. normal

As per central limit theorem, the mean of sampling distribution is approx. equal to the population means.

Equation of normal distribution

$$f(x/\mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}}\left(e^{-\frac{(x-\mu)^2}{2\sigma^2}}\right)$$

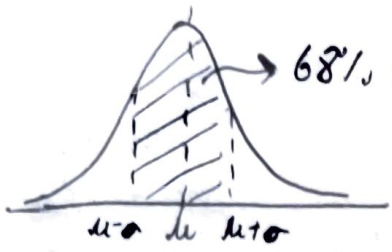
μ - mean

σ - standard deviation $\Rightarrow$ $\frac{\sigma}{\sqrt{n}}$ (for a sample) $\Rightarrow$ n is sample size

Normal distribution

- Area under a prob density func. gives the probability for the random variable to be in that range
- If population data available, upon taking random samples of equal size from data, the sample mean are approx. normally distributed.
- There is large prob. for the means to be around actual mean of the whole data, than be farther away.
- Normal dist. for higher standard deviation are flatter as compared for lower standard deviations.

A normal curve is symmetrical, unimodel and bell-shaped.



→ 68% of data falls within one std. devn. on either side of mean
ie b/n 'μ-σ' & 'μ+σ'

95% of data falls within two std. on either side of mean 'μ'
ie b/n 'μ-2σ' & 'μ+2σ'

Total area under standard normal curve is 1.

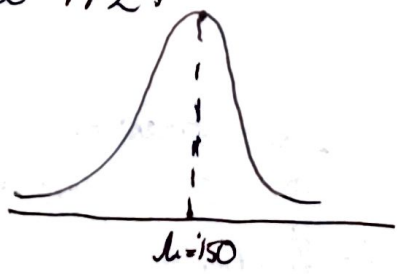
In std normal dist. the mode is the same as the mean, and the mean of a standard dist. is 0. Hence the mode is 0.

problem

Where would 172 fall on the distribution. How many people have cholesterol below 172.

$\mu = 150$

$\sigma = 15$



Ans

$\mu + Z \times \sigma = 172$

$150 + Z \times 15 = 172$

$Z = 1.47$

Sample statistic - The statistic of a group taken from a population

SD - It is amount of variation in the populatn data (σ)

o Z scores

The distance in terms of number of SD, the observed value is away from the mean is the standard score or the Z score

A positive z score indicates that the observed value is z standard deviations above the mean
Negative z score indicates value is below mean.

observed value = $\mu + z\sigma$

• The distribution once converted to z-score is always same as that of the shape of the original distribution

o Inferential statistics

• Making inferences about the population from the sample
• concluding whether a sample is significantly different from the population.

Inferential statistics is used to make inferences. We see the sample of data and decide its impact on the population data. So, ~~option b~~ it can be deduced that it used to draw conclusions beyond the immediate data available.

Inferential statistics comes into picture if the trend of our results goes the same way as our hypothesis; if our hypotheris are matching the problem at hand, we perform inferential statistics to check we should accept or reject the hypothesis.

statistic - A single measure of some attribute of a sample
population statistic - The statistic of entire population in context.

Confidence interval & Margin of error

Confidence interval in a given sample is an interval [LB, UB] with k% of confidence level

sample with mean $\bar{x}$ follows normal dist.

In sample $\mu_{sample} = \mu_{population}$

$$\sigma_{sample} = \frac{\sigma_{population}}{\sqrt{n}}$$

n → no. of sample

To calculate confidence level.

$$C.I. = \bar{x} \pm z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

$\bar{x}$ → sample mean

$z_{\alpha/2}$ → z value for desired confidence level α

σ → SD.

$z = 2$

$k\% = 95\%$

Problem

Calculate the 95% confidence interval for a sample mean of 40 and sample standard deviation of 40 with sample size equal to 100.

Ans. z for 95% CI = 1.96

$$CI = \left[\left\{40 - \left(1.96 * \frac{40}{\sqrt{10}}\right)\right\}, \left\{40 + \left(1.96 * \frac{40}{\sqrt{100}}\right)\right\}\right]$$

$$= [32.16, 47.84]$$

CI defines a range of values in which the population statistic may lie

Margin of error is half of the CI

Hypothesis testing

- Null hypothesis: can be that the sample statistic is to be equal to the population statistic or that the intervention doesn't bring any difference to the sample.
- Alternate hypothesis basically negates the null hypothesis or says that the intervention brings a significant difference to the sample, or that the sample is significantly different from the population.

eg: Introducing music in an effort to improve average mark of a class. If no improvement then its a null hypothesis otherwise if there has been improvement the its an alternate hypothesis.

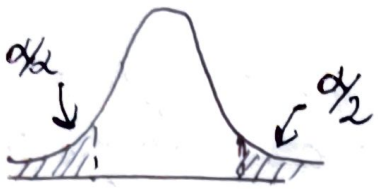
If Z-score < 0.5 ⇒ There is change due to intervention
rejection of null hypothesis.

- Null hypothesis is what we want to disapprove.
- In hypothesis testing we check sample against population statistic, check the sample after some intervention, to check a sample with another sample.

A critical value is the point on the scale of the test statistic beyond which we reject the null-hypothesis and is derived from the level of significance of the test.

In regards to this example, if we are only looking to see if there is any improvement then it is a directional (one tail) testing. If we're checking if the score decreases OR increases it is called non-directional (2 tail) testing.

In non-directional testing the α value is divided by 2



Errors

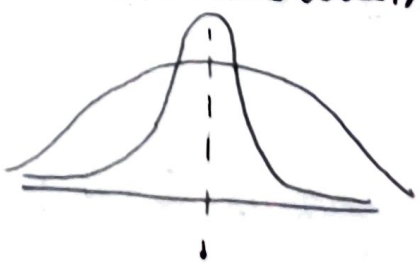
A hypothesis is not factual. It is highly dependent on the researcher.

Type 1 error (false +ve) – Rejecting the Null hypothesis when it's actually true

Type 2 error (false -ve) – failing to reject null hypothesis when it's actually false.

If a result is said to be significant at 5% level. It implies that the result would be unexpected if the null hypothesis were true

- T-tests
  - Very similar to Z scores
  - Uses sample SD to estimate population SD making it more prone to errors thus more wider tail compared to normal distn.



T value depends on degree of freedom.

Types: one tail T test
two tail T test

T test is used when we do not know the population SD for the variables hence we are more likely to have errors. Also, the number of samples used in t-test are less than 30. So T-distribution is more prone to errors

Example of demonstrating degree of freedom

- Selecting 4 numbers whose sum result is 10

We can select the first 3 numbers as we wish but 4th number is forced ~~by the~~ depending on the first 3. Hence in this case the degree of freedom is 3.

To read a t-table we require degree of freedom, significance level as well as we should also know the type of test ie either it is one tailed or two-tailed.

We failed to reject the null hypothesis if t-statistic is within t-critical value ie sample and population are similar to each other.

To calculate t statistic:

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}}$$

$s \rightarrow$ SD of sample

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

$n \rightarrow$ no. of instances in sample

Single tailed test only takes one side of the dist into consideration. It might be above or below the t-statistics. In two tailed tests we consider both the sides.

P-value for single tailed test is the prob. either above or below the t statistic

In one sample t test we take one sample and compare with the required result.

eg: To determine if assembly line makes laptops that weigh 5 kg. To test this hypothesis we collect sample of laptop computers from the assembly line, measure their weights, and compare the sample with a value of five using a one sample t-test

- 2 sample t-test

t value, t = $\frac{\text{difference b/n two values (mean)}}{\text{standard error}}$

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{S_1^2}{N_1} + \frac{S_2^2}{N_2}}}$$

$$\text{degree of freedom} = N_1 + N_2 - 2$$

- Paired t-test

In paired t-test we compare samples before and after intervention. So, null hypothesis will be the mean of samples differs significantly from the mean of sample after intervention

- Chi Squared test
  - for categorical variables

- $x_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$ O - Observed, E - Expected frequency

- To reject Null hypothesis we'll need significantly higher chi-squared value.
- It is unidirectional test.
- If calculated Chi squared statistic is below critical value it signifies that we failed to reject the Null hypothesis. ↑
- If p value > 0.05 <same>

- correlation
  - It is used to determine the relationship b/n two variables
    - denoted by r.
    - r ranges between -1 to 1. 0 represents no correlation

$$R = \frac{cov(X,Y)}{Sx, Sy}$$

↓ covariance shows how much there variables vary with each other. While the standard deviation shows how much there variables vary apart from each other

$$cov(X,Y) = \frac{\sum (x - \bar{x})(Y - \bar{Y})}{n-1}$$

$\bar{x}$: mean of 'x'; $\bar{Y}$: mean of random variable 'Y'; n: no of items in data set.

- Predictive ~~analysis~~ modelling - Using past data or historical data to predict some future values
  - Supervised learning (has a target variable)
    - Regression (continuous)
    - classification (discrete)
  - Unsupervised
    - Clustering

Stages of predictive modelling
- Problem defn.
- hypothesis generation - list down factors affecting our objective
- Data Extraction/collection from various sources
- Data exploration & transformation (insight to data to understand it clearly)
- Predictive modelling
- Model deployment/implementn.

Data Exploration/transformation steps
- Reading the data (RAW CSV, Sheet etc)
- Variable identification → its datatype, whether categorical or continuous (variable category) etc.
- Univariate analysis → analyse variables one by one using bar plots/histograms.
- bivariate → relatn. b/n two variable.
- Missing value treatement → identify and treat missing value using mean, mode, median etc.
- Outlier treatement

- Variable transformation → modify data to suit the algorithm or process.

Variable identification is the process of identifying which variables are dependent and which are independent, at same time they're identified as continuous and categorical.

Supervised learning require identification of dependent variable.

↳ The variable we're trying to predict

on the other hand indpdt help predict the depdt ones

categorical variable - discrete (stored as objects in pandas)

continuous " - any value b/n it's domain (real values)

↓

(stored as int or float in pandas)

dtypes - used to identify data.

↳ fixed integer values

Univariate analysis

- for continuous variable
  - Central tendency and dispersion
    - mean
    - median
    - SD
  - Distribution of variable
    - symmetric / Right skewed / Left skewed
  - Presence of missing variable
  - Presence of outliers.

◎ Methods :

- Tabular methods - best suited for mean, median, SD and missing values.
- Graphical methods - distribn of variables, presence of outliers.

Tabular methods
- describe()

Graphical methods
- histogram — y: frequency, x - element — pd.Dataframe.hist(), pd.Dataframe.boxplot()
- Boxplot - detectn of outliers.

- for categorical variable
  - count - absolute freq of each category
  - count% - proportion of diff. categories in the categorical variable expressed as %

◎ Methods
- Tabular - frequency → df['Sex'].value_counts()
- Graphical - bar plots → df['Sex'].value_counts().plot.bar()

for %

df['Sex'].value_counts()/len(df['Sex'])

(df['Sex'].value_counts()/len(df['Sex'])).plot.bar()

Bivariate Analysis

- Two variables are studied together for their empirical relationship
- It helps to make predictions
- It helps detect anomalies.

Types of variations :

- continuous - continuous variables

  method used :
  - Graphical method - Scatter plot.

  analysis test - correlation

$$r = \frac{Cov(x,y)}{\sqrt{S_x^2 * S_y^2}}$$

graphical method: boxplot

df.groupby('Sex')['Age'].mean()

mean of individual genders.

- categorical - continuous variables

  analysis test : 2 sample t test

$$t = \frac{\bar{y}_1 - \bar{y}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

$\bar{y}_1$ - mean of y_1
$\bar{y}_2$ - mean of y_2
s_1^2 - variance of y_1
s_2^2 - variance of y_2

- categorial - categorical variables

  graphical method

  analysis test → Two-way table → pd.crosstab( )

  → Chi-squared test

$$\chi^2 = \frac{\sum (O - E)}{E}$$

O → observed freq.
E → Expected frequencies.
χ^2 → statistic

chi_contingency (pd.crosstab(df['Sex'], df['Survived']))

Missing value Treatement

- Reasons for missing value
  - Non-response - omitting certain information by choice
  - Error in data collection
  - Error in reading data.

- Types
  - Missing Completely at Random (MCAR)
  - Missing at Random (MAR)
  - Missing not at Random (MNAR)

Consider an age, IQ table for examples

MCAR

Missing value have no relⁿ. to
- The variable in which missing value exists
- Other variables in the dataset.

eg: When Random IQs for various age groups are missing

MAR

no relⁿ to variable in which they exist, but have relⁿ. with other variables in which it exists.

eg: When only IQs of those below the age of 60 are missing

MNAR

Missing value have relⁿ. to variable in which it exists

eg: When IQs less than 100 are missing in IQ column.

Identifying missing values in variables

- describe() - only for continuous variables
- isnull() - for both continuous & categorical variables.

- Methods to deal with missing values
  - Imputation
  - Deletion → row-wise / column-wise

Imputation:
- categorical → mode → classification model
- continuous → mean / median → regression model

Outlier Treatment

Reasons
- Data entry error
- Measurement errors
- processing errors
- change in underlying population

- Types
  - Univariate outliers – use boxplot to identify
  - Bivariate outliers – use scatterplot to identify

(graphical method)

Formula methods

Outlier is defined as any value which is less than $Q_1 - 1.5 * IQR$ or greater than $Q_3 + 1.5 * IQR$

Treating
- Deleting observations
- Transforming and binning values
- Impute outlier similar to missing values
- Treat them separately

Variable Transformation

It is process by which -

- We replace a variable with some function of that variable. eg; replacing a variable with its logarithm
- We change the distrn. or reln. of a variable with others.

Purpose of variable transformation -

- change the scale of variable.
- convert non-linear reln. to linear one
- create symmetric distn. from skewed distn.

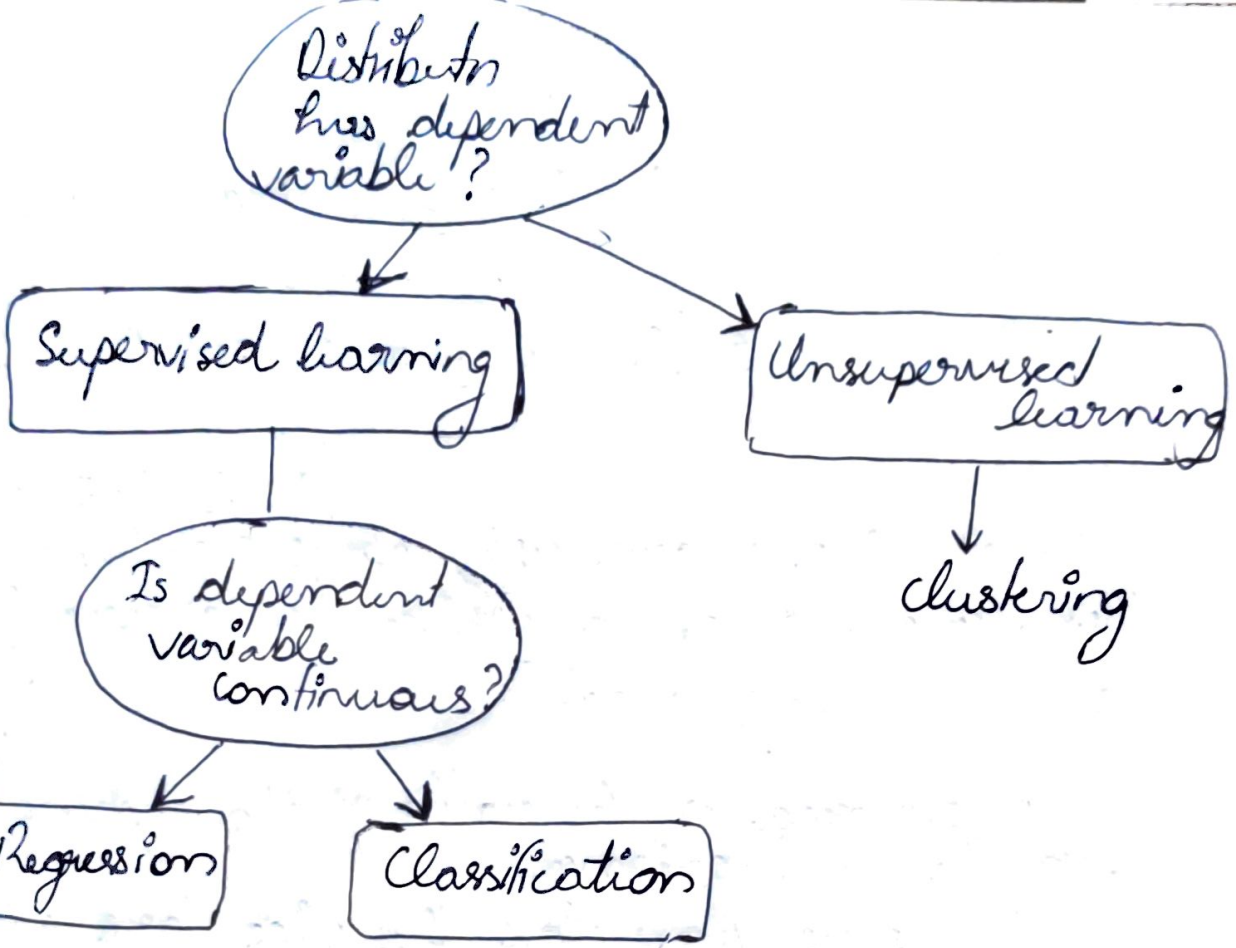
right skewed.

Common methods:

- logarithm
  - reduces right skewedness of variable.
- square root
  - reduces right skewedness with +ve values only
- cube root
  - reduces right skewedness for both ~~left~~ +ve or -ve values
- binning - converts continuous variables to categorical variables.

Model building

Algorithm selection ⇒ Training model ⇒ Prediction/ Scoring



--> Linear regression

$\hat{y} = \theta_0 + \theta_1 x$

approximation value

θ_0 - intercept [$y = mx + c$]
θ_1 - slope
x - independent variable
y - target variable

θ_0, θ_1 value found such that errors are minimum

cost function : Squared error

$(\hat{y} - y)^2$

Mathematically -> $\frac{1}{2n}(\theta_0 + \theta_1 x - y)^2$

reduces computation & mean error

Gradient Descent algorithm

find θ_0, θ_1 using gradient

Let c be cost function,

$$c = \frac{1}{2n}\sum(\hat{y} - y)^2$$

$$C = \frac{1}{2n}\sum (\theta_0 + \theta_1 x - y)^2$$

for eq. $\hat{y} = \theta_1 x$

$$\theta_1 = \theta_1 - \frac{\delta C}{\delta \theta_1}$$

$$\frac{\delta C}{\delta \theta_0} = \frac{1}{n}\sum (\theta_0 + \theta_1 x - y), \text{ gradient w.r.t } \theta_0$$

$$\frac{\delta C}{\delta \theta_1} = \frac{1}{n}\sum (\theta_0 + \theta_1 x - y)x \text{ ; w.r.t } \theta_1$$

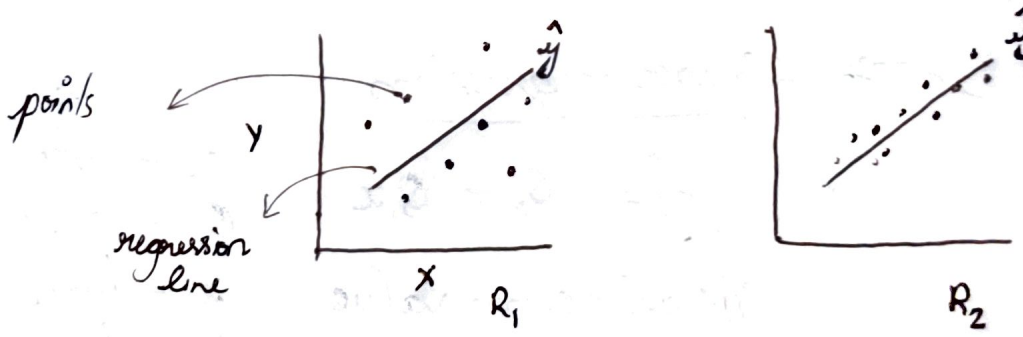
if slope $\uparrow$ $\theta_1 \downarrow$ otherwise $\uparrow$ till it reaches an optimum value

if there are $\theta_0, \theta_1, \theta_2 \ldots$ use gradient diffn similar to Gauss Seidal iteration (34)

- Having optimised model using above algorithm, we evaluate its performance, for which the generally used metric is 'r-squared'

R-squared is the ratio of the explained variance upon the total variance.

It tells us how close the points are wrt the line '$\hat{y}$'



$R_2 > R_1$

- Mathematical form of R-squared

$$r^2 = \frac{\sum (\hat{y} - \bar{y})^2}{\sum (y - \bar{y})^2} = \frac{SSR}{SST}$$

Squared sum of regression / Squared sum of total variance

$\bar{y}$ → actual mean.

$$= 1 - \frac{SSE}{SST}$$

Sum of squared error $= \sum (\hat{y} - y)^2$

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

Evaluation metrices

- Root mean square Error (RMSE)

Better model would have lower RMSE

$$\text{Mean square error} = \frac{\Sigma(y-\hat{y})^2}{n}$$

$$\text{RMSE} = \sqrt{\frac{\Sigma(y-\hat{y})^2}{n}}$$

In a linear regression problem, we are using "R-squared" to measure goodness-of-fit. We add a feature in linear regression model and retrain the same model. But, Individually R squared cannot tell about variable importance. We can't say anything about it right now. This is because R squared value never decreases with addn. of a new feature. It either increases or remains the same. So, R squared can't tell about variable importance.

Linear regression is used for regression problems. So, we can not use linear regression when the dependent variable is categorical.

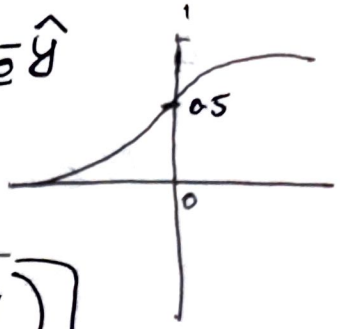
p value and t-statistic does not tell about the relⁿ. b/n two variables. They just tell us whether or not they are statistically similar. Correlation coefficient on the other hand tells us how strongly two variable are related to each other.

- Logistic Regression

result is value b/n 0 & 1

Sigmoid function used

$\hat{y}$, $\frac{1}{1+e^{-\hat{y}}}$



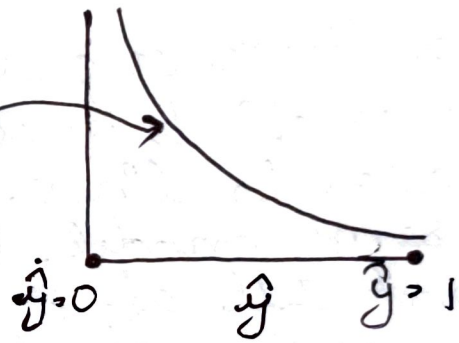
cost function,

$$C = \begin{cases} -\log(\hat{y}) & \text{when } y=1 \\ -\log(1-\hat{y}) & \text{when } y=0 \end{cases}$$

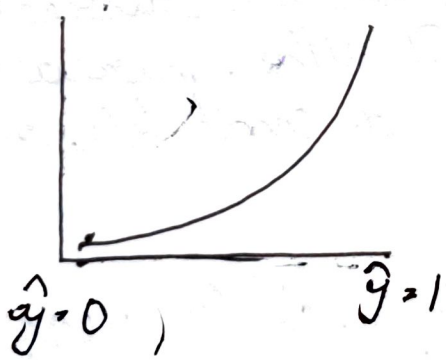
limits $\hookrightarrow [0,1]$

$$\boxed{C = -y(\log(\hat{y})) - (1-y)\log(1-\hat{y})}$$

When $y=1$, $C = -\log(\hat{y})$



$y=0$



Optimisation

Using gradient

$C = -y(\log(\hat{y})) - (1-y)\log(1-\hat{y})$

Where $\hat{y} = \theta_0 + \theta_1 C$

$$\theta_0 = \theta_0 - \frac{\delta C}{\delta \theta_0}$$

$$\theta_1 = \theta_1 - \frac{\delta C}{\delta \theta_1}$$

Evaluation metric

Using actual positive and predicted positive, and actual negative predicted negative table, methods such as AUC-ROC, Accuracy, Loglass may be applied for classification problems.

Logistic regression can be used on 3-class classification problem.

Decision Tree

- Each node is a yes/no decision
- Depth of node signifies no. of decisions/nodes to be traversed before reaching conclusion (Terminal node).
- Root node contain entire set of objects
- With each decision made the amounts of objects at a node is split and relevant object set is passed further down.

Decision tree splitting is done by asking generic question, generic questions mean those questions relevant to splitting the data most efficiently

Gini Index → measure of quality of split

- Only applicable for 2 class classification problem.
- Bigger the gini index, better is the split.
- For a sub node, calculates the sum of scares of proportions of two classes prese it in node

To calc gini index of a split, we take a weighted sum of ginis of the two subnodes, where the weights are the proportions of the parent node

present in a submode.

- Entropy value lies b/n 0 and 1
- We can use decision tree for both classification and regression problems. In classification, we take the mode of the values at the leaf node to make predictions. Whereas, in regression we take the mean of values at the leaf node.

Unsupervised learning

- Clustering - dividing the population or data-points into a number of groups such that data points in same group are more similar to other points in the same group than other points in other groups.

  The aim is to segregate groups with similar traits and assign them into clusters.

- Types of clustering
  - Hard clustering: Where datapoint belongs to one cluster
  - Soft clustering: Where one data point can be in multiple clusters with some probabilities

A popular hard-clustering algorithm is k-means

Steps:

1) Decide no. of clusters (say, depending no. of datapoints)
2) Randomly assign each point to a cluster.

3) Calculate the cluster centroid

↓

average of points along the axis.

4) Calculate the distance of distance of each point from the cluster centroid and assign a point to cluster for which distance is minimum. The recompute the cluster centroid.

Stopping criteria :

-> No. of iterations - no. of times we calculate cluster centroids

-> When cluster centroids do not change

-> Points do not change their cluster.

k-means uses uclidean distance and minimize it to assign an object to a cluster

---

Exporting Pandas DataFrame to a csv file.

-> df.to_csv(r'Path where you want to store csv file\filename.csv')

Creating a dataframe

```
-> cars = {'Brand':['Honda','Toyota','Ford','Audi'],
           'Price':[22000,25000,27000,35000]
          }
df = DataFrame(cars, columns=['Brand','Price'])
print(df)

<or>

df = pd.DataFrame(cars)
```

Iterating through records in a dataframe

- Inorder to iterate over rows, we apply a iterrows() function; it returns index value along with a series containing the data in each row.

index → pandas series

```
for i, j in data.iterrows():
    print(j)
    print()
```

output

```
Honda
Price      22000
Brand      Toyota
Price      25000
```

// ... and so on

- iteritems() function iterates over each column as key, value pair with label as key and column value as a series object.

```
for key, value in df.iteritems():
    print (key, value)
    print
```

output

```
Brand  0  Honda
       1  Toyota
       2  Ford
       3  Audi
       ....

Price  0  22000
       1  25000
       2  27000
       3  35000
```

- itertuples() return a tuple for each row in dataframe. The first element of the tuple will be the rows corresponding index value, while remaining values are the row values.

```
for i in df.itertuples():
    print(i)
```

output:

```
Pandas (Index=0,        ='Honda',
                Price = 22000)
Pandas (Index=1, Brand = 'Toyota', Price=25000)
    :
Pandas(Index=3, Brand = 'Audi', Price = 35000).
```

Iterating over columns:

```
colmn = list(df)
for i in colmn:
    print(df[i][2]) # printing a third
                      element of colmn.
```