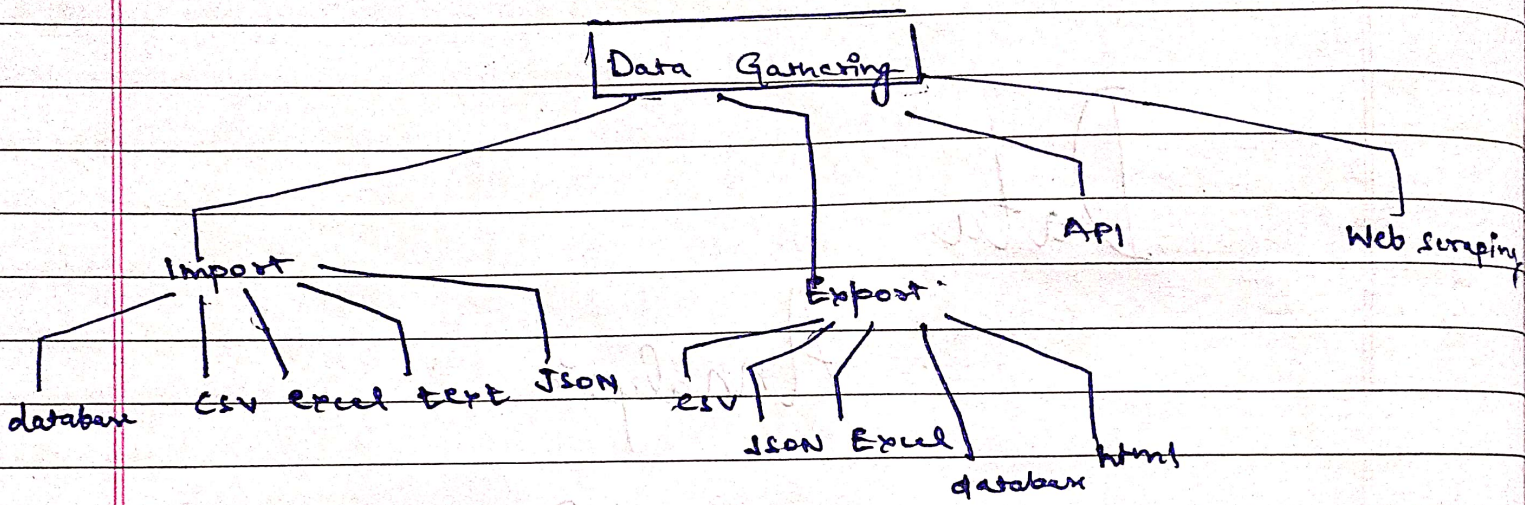
Data Annalysis Process

RANKA
DATE / /
PAGE

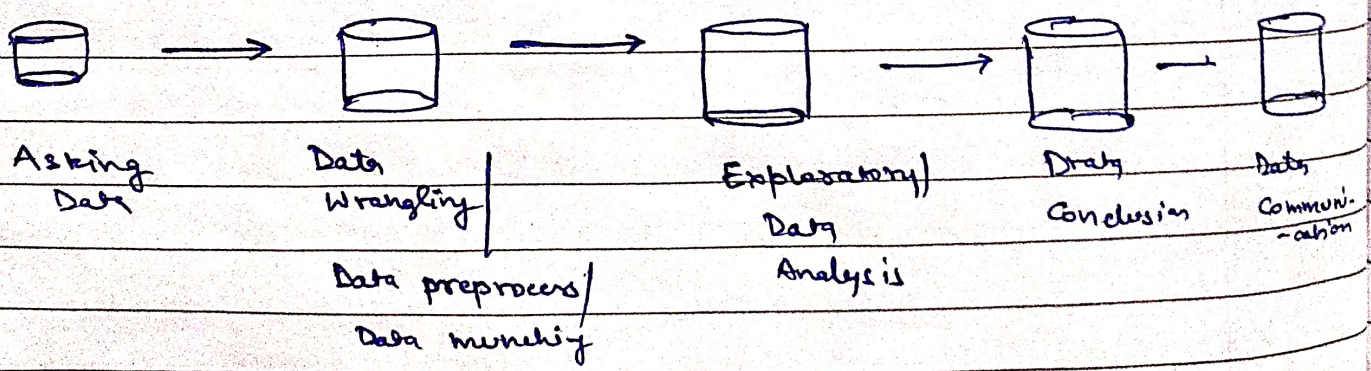
Data Analysis process

1) Data Gathering.
2) Data Cleaning
3) EDA.



Data Analysis is a process of inspecting, cleansing, transforming and modelling data with a goal to discover useful information.

Data Analysis process



RANKA
DATE / /
PAGE

Way 1 : get the data

Step 1 : Asking Qu'

Way 2 → problem statement is given

Ques1 What feature will contribute to my analysis

Ques2 What feature are not important for my analysis

Ques3 Which of feature have strong correlation?

Ques4 Do I need data preprocessing

Ques5 What kind of feature manipulation/ engineering is required?

How can I ask better question

1) Subject matter expertise
2) Experience

Step2 Data Wrangling / Munging :

process of transforming and mapping data from one "raw" data form into another with an intent of making it appropriate.

1) Gathering data → [CSV, API, Web scrapping, Databases]
2) Assessing data [shape, info, is_unique, describe]
3) Cleaning data : [

Ass data : 1) Find no. of rows / column (shape)
2) Data type of various columns (info())
3) Checking for missing values (info())
4) Check for duplicate data (is_unique)
5) Memory occupied by the dataset (info)
6) High level mathematical overview of the data (describe())

RANKA
DATE / /
PAGE

Cleaning data

1) Missing Data (eg. mean)
2) Remove duplicate data (drop_duplicate)
3) Incorrect data type (astype)

Step 3 Exploratory Data Analysis

Explore	Augment (Feature Engineering)
1) Find Correlation & Covariance	1) Removing outlier
2) Doing Univariate and multivariate analysis	2) Merging Dataframe
3) Plotting graphs (data visualisation)	3) Adding new Column

Step 4 Drawing Conclusion

Take a small sample try to conclude to entire (Inferential Stats)

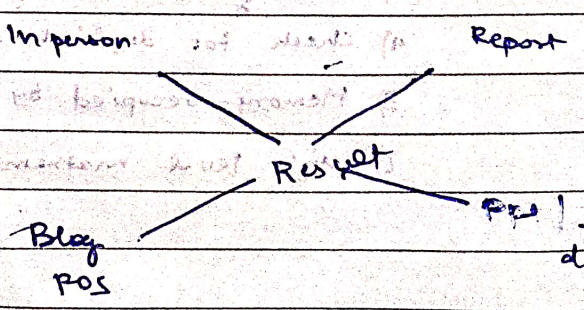
Machine learning (prediction) — Data Science — Conclusion Conclude

Descriptive Statistics

Descriptive Stats

1) Is Rohit Sharma a better batsman in 2nd innings (IPL)
2) Does being a female increase your chance of survival (Titanic)
3) Is delhi the most costly place to eating out (Zomato)

Step 5 Communicating Result / Data story telling



CSV — comma seprated value

tsv — tab seprated value

document read-csv pandas

1) df = pd.read-csv('name') → Open a local csv file.

2) Opening a csv file from a URL

```
import requests.
from io import StringIO.
url = ''
headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14;
                          rv:66.0) Gecko/20100101 Firefox/
                                                66.0"}
req = requests.get(url, headers = headers)
data = StringIO(req.text)
pd.read-csv(data)
```

3) Sep parameter

pd.read_csv('file.tsv', sep='\t', names=[n1, n2, n3, ...])

अगर सारे column में names नहीं दिए हैं

4) Index-col parameter:

pd.read-csv('file', index-col = 'column-name')

Use data Index

5) Header parameter:

pd.read-csv('test.csv', header = 1)

अगर column name की यह नहीं row में है

6) Use-col parameter.

, अगर सभी के बजाए column ही चाहिए.

pd.read-csv('file', use-cols = ['enrol', 'gender', 'edu'])

7) Squeeze parameter

pd.read-csv('file', usecols = ['gender'], squeeze = True)

यह एक column होगा तो series object होगा

8) skiprows / nrows Parameter

pd.read_csv('file', skiprows = [0,1]) → nrows = 100

अगर कोई row skip करनी है

nrows → अगर कितने rows load करनी है

9) Encoding parameter

अगर error है Unicode DecodeError

pd.read_csv('zomato.csv', encoding = 'latin-1')

10) skip bad lines :-

pd.read_csv('file', sep = ',', error_bad_lines = False)

11) dtypes :-

अगर हमें column की datatype change करनी है

pd.read_csv('file', dtype = {'target' = int})

12) Handling Dates

अगर default string dates को datetime dtype में change करनी है।

pd.read_csv('file', parse_dates = ['date']).info()

अगर date multiple columns में है। |month | year|

pd.read_csv('file', parse_date = [[1,3]])

Imp

13) Converters

```
def rename(name):
    if name == 'Royal Challengers Bangalore':
        return 'RCB'
    else:
        return name
pd.read_csv('file', converters = {'team': rename})
```

(team column name)

RANKA
DATE / /
PAGE

14) na_values parameter

pd.read_csv('file', na_values=['undefined'])

इसमें कोई string value होगी undefined, - या null value में convert होगी।

15) Load a huge dataset in chunks

dfs = pd.read_csv('avg', chunksize = 5000) → 19000

for chunks in dfs:
    [operations]
    print(chunk.shape)

Excel

pd.read_excel('output.xlsx')

pd.read_excel('output_file', sheet_name='Sheet_name_2')

Data from text

pd.read_csv('file', sep='\t')

JSON | SQL

JSON → Javascript Object Notation
(universally accepted format)

SQL → structure query language

→ documentation: pandas read_json
pandas read_sql query.

JSON

1) from Local machine
data = pd.read_json('train.json')

2) from url

pd.read_json('url')

SQL

World-Cities, Pop, Lang, Rank, (SQL create)

Step1 : download dataset

Step2 : download Xamp. Start → Apache, MySQL

↓

localhost/phpmyadmin

1) Create a new database
2) Upload world sql file.

Step3 : !pip install mysql.connector.

Step4 : import mysql.connector.

conn = mysql.connector.connect(host='localhost', user='root', password='', database='world')

data = pd.read_sql_query("SELECT * FROM city", conn)

Pandas Export

- to_csv
- to excel
- to html
- to json
- to sql

→ Google colab

CSV

```
temp_df = df.groupby('batsman')['batsman_runs'].sum().
          reset_index()

temp_df.to_csv('batsman_runs.csv', index = False)
```

to_Excel

1) `temp_df.to_excel("batsman_runs.xlsx")`

2) Save to some another sheet.

```
temp_df.to_excel("batsman_runs.xlsx", sheet_name
                 = 'batsman_runs')
```

3) Make two diffren excel sheet in same file

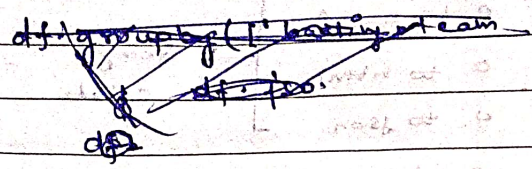
```
with pd.ExcelWriter('output.xlsx') as writer:
    temp_df.to_excel(writer, sheet_name='One')
    temp_df2.to_excel(writer, sheet_name='Two')
```

to_html

```
df.query('batsman_runs == 6').pivot_table(index='over', columns
    ='balls', values='batsman_runs', aggfunc='count').to_html
    ('Size_heatmap.html')
```

to_json

```
df.to_json('file')
```



to_sql

Step1 Create a new database on using Xampp

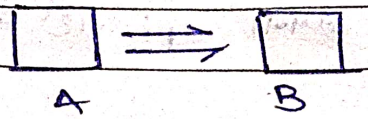
Step2

```
import pymysql.
from sqlalchemy import create_engine

engine = create_engine("mysql+pymysql://root:
    @localhost/ipl")

# {root}:{password}@{url}/{database}
df.to_sql('ipl_delivery', conn = engine, if_exist='app.
    end')
```

↑ Table name

Indirect data → इसमें case में हम दूसरे की website से Api से data लाएंगे.

API →



1) API pipelines जो data को A से B तक ले जाती है or

2) API दो software के बीच में बात कराती है

website rapid api

Step-1 tmdb api → google

Step-2 find url. → <api key>
- create account
profile > setting > api key.

Step-3 paste full website on json viewer.

Step-4
pages → 1 — 458
[ ]
page - 428.
total movies - 8551

Step-5 code.

id
title
release date
popularity
vote_average
(8851, 5)

```
import pandas
import request
.
df = pd.DataFrame()
for i in range (1,429):
    response = request.get('<link>
    .format(i))
temp_df = pd.Dataframe(response.json()['results']) [['id', 'title', ...]]
df = df.append(temp_df, ignore_index=True)
df.shape
df.to_csv('movies.csv')
```

Web scraping

अगर कोई website अपनी api नहीं दे रही हो।

```
import pandas as pd
import requests
from bs4 import BeautifulSoup.
```

beautiful soup 4. →

```
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT ...)'}
webpage = requests.get('https://www.ambitionbox.com/list-of-companies?page=1', headers=headers).text

soup = BeautifulSoup(webpage, 'lxml')

Company = soup.find_all('div', class_='company-content-wrapper')

name, rating, reviews = [], [], []
→ ctype, hq, old, employees = [], [], []
for i in company:
    name.append(i.find('h2').text.strip())
    rating.append(i.find('p', class_='rating').text.strip())
    reviews.append(i.find('a', class_='reviews-count').text.strip())

    ctype.append(i.find_all('p', class_='info entity')[0].text.strip())
    "
    "
    "

d = {'name': name, 'rating': rating, 'reviews': reviews,
     'type': ctype, 'hq': hq, 'old': old, 'employees': empl}
df = pd.DataFrame(d)
df
```