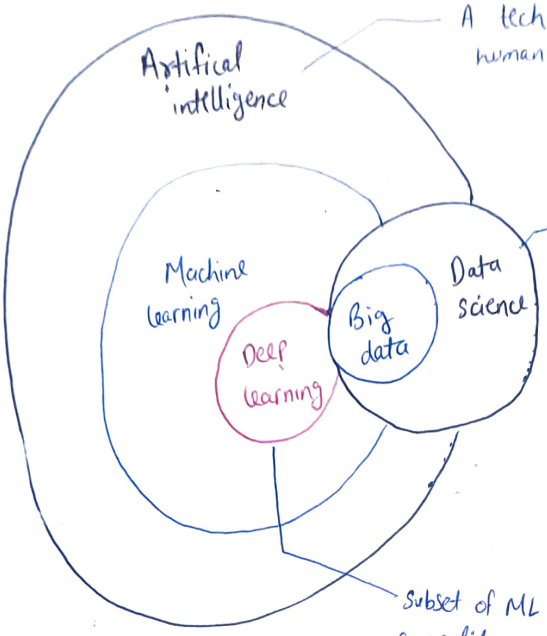
Machine Learning (Udemy)

Machine learning

is study of computer algorithms that improve automatically through experience.



A technique which enables machines to mimic human behaviour

inter-disciplinary field that uses scientific methods, processes, algo & systems to extract knowledge & insights from many structural & unstructured data.

Subset of ML which make computation of multi-layer neural network feasible

Types of Machine learning

Supervised learning

→ Classification
- Fraud detection
- Email spam detection
- Diagnostics
- Image classification

→ Regression
- Risk assessment
- Score prediction

for classification
→ Naive Bayes → SVM
→ K-Nearest neighbour

for regression
→ Decision tree
→ Linear Regression
→ Logistic Regression

Unsupervised learning

→ Dimensionality Reduction
- Text mining
- face recognition
- Big Data visualization
- Image Recognition

→ clustering
- Biology
- city planning
- Targeted marketing

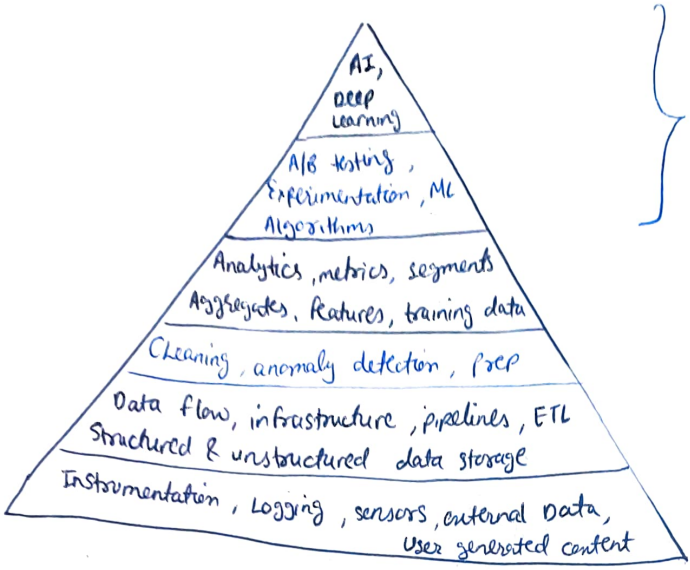
for clustering
→ K means
→ Mean shift
→ K-Medoids

Dimensionality Reduction
→ Principal Component Analysis (PCA)
→ feature selection
→ Linear Discriminant Analysis (LDA)

Reinforcement Learning

- Gaming
- finance sector
- Manufacturing
- Robotic navigation
- Inventory Management

Data Science hierarchy of needs



Section 2 : Predict Mokie Box office Revenue with Linear Regression

- Steps to Solve a problem by Data Science
  1) formulate Question
  2) Gather Data
  3) Clean Data
  4) Explore & Visualize
  5) Train algorithm
  6) Evaluate.

1) Formulate Question → make well defined question

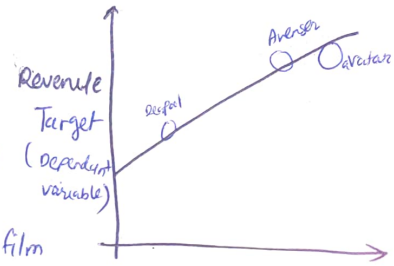
eg • how much money our film make? Poor question

• how much revenue our film make ? (little better)

But every input for film relates to money used to make film

So best question

→ "Can we use movie budgets to predict movie revenue ?



2) Gather data

for this we want budget (in $) & revenue (in $)

find some sites where you can get this

3) Clean data

here for eg: we have some films with revenue 0$, their are either one to be released or never released but made.

so remove such problematic data.

use excel to clean this data.

also if data is like $43,000,000 but to analyze we want 43,000,000 so clear $ sign from excel (format data) (save this as 'cost_revenue_clean.csv')

4) Explore & Visualize

can use "https://jupyter.org/try" / google collab. / vs code → notebook.

To explore data → pandas , to visualize data → matplotlib

Code

```
import pandas
from pandas import DataFrame
import matplotlib.pyplot as plt
```
// Data frame is two-dimensional data structure.

```
data = pandas.read_csv('cost_revenue_clean.csv')
data.describe()
```
// for getting count, mean etc details

also eg 5.034000e+03 → scientific notation means 5034 (5.034 x 10^3)

```
X = DataFrame(data, columns=['Production_budget_usd'])
y = DataFrame(data, columns=['worldwide_gross_usd'])
```
→ from csv

```
plt.figure(figsize=(10,6))
plt.scatter(X,y)
plt.title('Film cost vs Global revenue')
plt.xlabel('Production Budget $')
plt.ylabel('world wide Gross $')
plt.ylim(0, 3000000000)
plt.xlim(0, 450000000)
plt.show()
```
// to adjust size

// graph is betn X & y (must to include)

↑ Scatter graph not line graph (for line graph use .plot)

// to range x & y according to data.

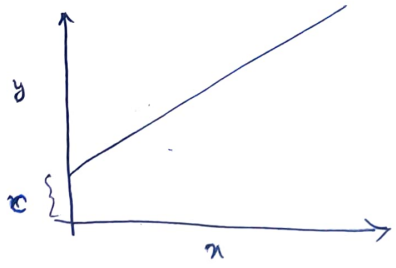
// to show data (must to include)

// save file

Linear Regression

- Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and other is considered to be dependent variable.

So we try to make

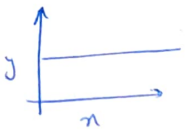


$y = mx + c$ for this

m & c = parameter

→ more the m, stronger the relation

as



$y = 5$

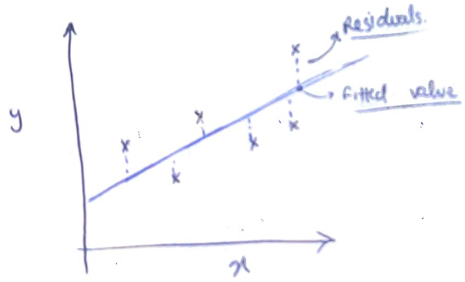
no relation bth x & y

$c = \theta_0$, $m = \theta_1$, $y' = h\theta(x)$

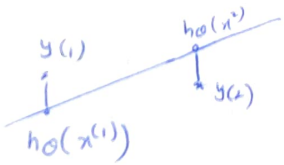
So

$$h_\theta(x) = \theta_0 + \theta_1 x$$

- now for given data many lines can be made we consider line which generate minimal residual



Residual =



So $y^{(1)} - h_\theta(x^{(1)})$ but for some like $y^{(2)} - h_\theta(x^2)$ it will be negative

So we square it

So we have to minimize

$$: (y^1 - h_\theta(x^{(1)}))^2 + (y^2 - h_\theta(x^{(2)}))^2 + (y^3 - h_\theta(x^{(3)}))^2 + \text{- - - - - -}$$

minimize : $\sum_{i=1}^{n} (y^{(i)} - h_\theta(x^{(i)}))^2$

Residual sum of Squares (RSS)

- For implementing linear regression we will use Scikit learn

(for fresh notebook, Run all)
↑ from cell.

- Jupyter Tip.
  Code ▾
  convert to markdown, to write explanation of code

▶ from sklearn.linear_model import LinearRegression

to use (after X, y dataframe generation etc task)

▶ regression = LinearRegression()
  regression.fit(X, y)

in fit X, y are 2d np arrays

** (if errors → see documentation of that fn) by hover on it

- for comment #theta_1

▶ regression.coef_ # theta 1 value, slope

▶ regression.intercept_ # theta 0 value, intercept

if slope = 3 means for each dollar in budget we get 3 dollars in revenue

to plot this on graph
copy paste old code.

▶ plt.figure(figsize=(10,6))
  plt.scatter(X, y, alpha=0.3) → opacity of dots
  plt.plot(X, regression.predict(X), color='red', linewidth=4) → to make line
  ⋮ } axis labels etc.
  plt.show().

So let your budget = P.
So how much revenue you get ⇒ revenue = $\theta_0 + \theta_1 P$
(regression.intercept_ → θ_0, regression.coef_ → θ_1)

- how good your analysis was.

"Goodness of fit" of linear regression model attempts to get at perhaps surprisingly tricky issue of how well a model fits a given set of data, or how well it predicts future set of observations".
represented as r^2 or R^2.

→ regression.score(X, y) to get r^2
  0.54 - - -
  means nearly 55% accuracy of prediction