(100 Days of ML) → 132 videos

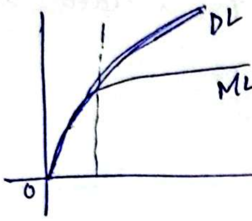
① **Lecture # 01 (video – 1)** → 20 min.

→ Intro to ML only.

Explicit Programming → code for specific logic/program.

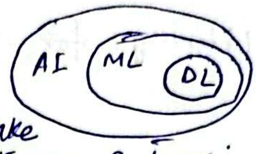
② **Lecture # 02 (video – 2)** → 16 min.

→ AI vs ML vs DL.



ML → Statistical technique to identify Data patterns.

DL ↳ Big Data (Large)



DL → Make its own features.

ML → You have to provide or make affecting features.

③ **Lecture # 03 (video - 3)** → 27 min.

→ Types of ML.

Supervised | Unsupervised (only input.) | Semi-Supervised | Reinforcement.

Supervised → input/output Both are given.
- RC
- → Regression
- → Classification

Unsupervised (CDAA)
- → Clustering
- → Dimensnality reduction.
- → Anamoly + Detection.
- ↳ Association (Rule learning).

Data
- Numerical
  - Age
  - cgpa etc
- Categorical
  - Gender
  - Nation

Clustering ⇒ Clusters Formation.

Dimenality reduction ⇒ Remove extra column (Does affect the output)
↳ Majority used in PCA, Feature Extraction, TSNE etc.
→ Outlier removal.

Regression ⇒ Target column should be numerical.

Classification ⇒ Target column should be categorical.

④ **Lecture # 04 (video - 4)** → 11 min

→ Batch vs Online ML.

Batch ML → Train model on localhost and then test model is deployed on server.
Disadvantage, one model is trained, not up-to-date.

Online ML → Training model within period of time (efficient way)
up-to-date model.

(5) Lecture # 5 (video-5) → 20 min.

(12)

→ Online vs Offline learning.

Online learning → Data → Model → server → New Data → train → Pred team

Mini- Batch learning

Concept Drift:

→ Nature of problem changes volatility.

→ River → Library for online streaming. (Train data up-To-date)

Vowpal Wabbit

↓

github repo.

(6) Lecture # 6 (video-6) → 17 min.

→ Instance-Based Vs Model-Base learning.

lazy learning ← Instance → eg KNN etc. Instance learning. (when new query comes it check shortest distance and give instance discussion.

Model-Based ⇒ learning without new query point. eg. LR, LogR, Decision Tree etc.

(7) Lecture # 07 (video-7) → 24 min

→ Challenges:

1. Data collection → Fetch data from API → Web scraping.
2. Insufficient/Labelled Data. → Unreasonable effectiveness of Data → In huge amount of data, algorithm doesn't matter.
3. Non-representative Data. → Sampling noise/Bias
4. Poor Quality Data → outliers, empty values etc.
5. Irrevalant Features, ↳ irrevalent feature in the Data.
6. Over fitting (7) Underfitting.
8. Software Integration
9. Offline Learning/Deployment
10. Cost Involved.

(8) Lecture # 8 (video-8) → 30 min

→ Application of Machine Learning

① Retails Sector ② Banking & Finance. ③ Transport → OLA ④ Manufacturing
⑤ Consumer Internet – Twitter etc.

⑨ **Lecture # 9 (video-9)** → 25 min.

→ Development life cycle – (ML). (MLDLC)

① Frame the problem
② Gathering Data
③ Data Preprocessing.
↳ Remove duplicates etc.
④ Exploratory Data Analysis
↳ Visualization
↳ Univariate
↳ Bivariate
↳ Multi-variate.
⑤ Feature Engineering and Selection
⑥ Model Training, Evaluation, selection
⑦ Model Deployment
↳ Heroku
↳ AWS, GCP
↳ Azure
↳ etc.
⑧ Testing
⑨ Optimize

⑩ **Lecture # 10 (video-10)** → 27 min

→ DE Vs DA Vs DS vs MLE Jobs roles.

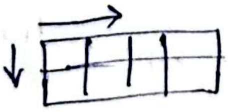
Data Engineer → Scrape Data (Hard field). → Gather data.
Data Analyst → Cleaning data → Past (see Past Data)
Data Scientist → Full-stack. → (Future Data → prediction).
↳ Big company → Model Deployment etc.

ML Engineer → Deployment of Model. / optimize / maintainance

⑪ **Lecture # 11 (video-11)** → 42 min.

→ what is Tensor? → to store data.
↳ vectors, matrixs etc.
→ Containers for storing numbers.
→ ndarray are called Tensor.
→ Tensor → no of axes = rank = no of dimensions

→ 1D Tensor = Vector.
→ 2D Tensor = Matrixes
→ ND Tensor
→ Shape, size of tensor.



→ ndimension or rank = ②
↳ shape → 2, 4, size = 2×4 = $\boxed{8}$

① 1D Tensor → used in ML types like classification etc.
② 2D Tensor → also used as in same senario as ①.
③ 3D Tensor → used in NLP mainly, Time-series Data etc.
④ 4D Tensor → used in images, computer vision, Deep Learning etc.
⑤ 5D Tensor → used in video frames, etc.

⑫ Lecture # 12 (video-12) → 37 min.

→ Setting up tools.

→ Creating virtual environment :

Conda create --name environment name}

Conda activate environment name

conda install -c anaconda jupyter.

deactivate enviroment → deactivate

Remove env. → conda remove --name environment name

To upload Big data use command on Google colab (GPU, CPU, TPU) ↳ can used for deeplearning.

!mkdir -p ~/.kaggle

!cp kaggle.json ~/.kaggle/

Before this make api token from kaggle.

Then Just copy api command and paste (all done).

⑬ Lecture # 13 (video-13) → 30 min.

→ End to End Project.

Placement Y/N → Logistic Regression.

⑭ Lecture # 14 (video-14) → 22 min.

→ Frame of ML Problem.

Churn rate = ~~Total uses~~ out of ~~time~~ Total users, leave users in given period of time.

⑮ Lecture # 15 (video-15) → 36 min.

→ Data Gathering.

(CSV) | Json/sql | fetch API | Web/scrapping

CSV Gathering :

These are parameters of pandas.

pd.read_csv(" ", sep=' ' etc)

Here you can pass parameters.

sep=',' → used to distinguish object.

name=[] → passes array in the place of headings.

index_col=' ' → make another column as index-col.

header='1' → 2nd row is header so used this code.

more parameters ⇒ usecols, sequeeze, skiprows/nrows etc

⑯ Lecture # 16 (video-16) → 17 min

→ Working with JSON/SQL
→ Working with SQL.

⑰ Lecture # 17 (video-17) → 23 min

→ Working with API.
→ RapidAPI. → used to make dataset.

⑱ Lecture # 18 (video-18) → 37 min

→ Fetching data using web scrapping.

⑲ Lecture # 19 (video-19) → 15 min.

→ Understanding your data.

1- How big is the data? → df.~~sample(s)~~ shape
2. How does the data look like? → df.~~info()~~ sample()
3- What is the data type of cols? → df.info()
4- Are there any missing value? → df.isnull().sum().
5- How does the data looks like? → df.describe()
6- Are there are duplicate values? → df.duplicated().sum().
7- How the correlation between cols? → df.corr().

⑳ Lecture # 20 (video-20) → 30 min.

→ EDA using Univariate Analysis.

Univariate Analysis

Categorical data
↳ Countplot -
↳ Piechart -

Numerical data
↳ Histogram -
↳ Distplot - (Kernel Density Function)
↳ Boxplot -

㉑ Lecture # 21 (video-21) → 39 min.

→ Bivariate and Multivariate Analysis.

Generally used for both Bi- and Multi-variate.

→ Scatter plot (Numerical- Numerical). sns.scatter
→ Bar plot (Numerical- Categorical) sns.barplot
→ Box plot (Numerical- Categorical) sns.boxplot
→ Dist plot (Numerical - Categorical). sns.distplot
→ HeatMap (Categorical - Categorical) sns.heatmap
→ ClusterMap (Categorical - Categorical). ~~pd.crosstab~~ sns.clustermap
→ Pairplot
→ lineplot (Numerical - Numerical) → x quantity should be time or number based.

Types of Encoding:

↳ ordinal Encoding → used with ordinal data
↳ One Hot Encoding → used with nomial data (the values of the column does not have relationship)

(22) Lecture # 22 (video-22) — 13 min.

→ Pandas Profiling.

(23) Lecture # 23 (video-23) — 25 min.

→ Feature Engineering.

- Feature Transformation
  - → Missing values Imputation ← Simple Imputator
  - → Handling Categorical Features → One-Hot code.
  - → Outlier Detection.
  - → Feature Scaling → Min-Max scaling.
    - ↳ Standardization.
    - ↳ Normalization
    - ↳ Mean absolute scaling.
- Feature Construction
  - (eg, making new column from two colums etc)
- Feature Selection
- Feature Extraction
  - ↳ TSNE
  - ↳ LDA
  - ↳ PCA etc.

(24) Lecture # 24 (video-24) — 32 min

→ Standardization (Feature Transformation) → $\boxed{\text{Feature Scaling}}$

Types of feature Scaling.

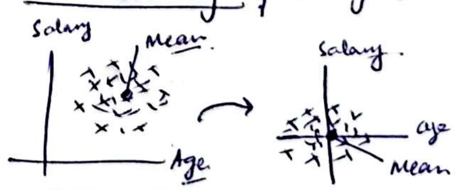
- ↳ Standardization
- ↳ Normalization
  - → Min-Max scales.
  - → Roboust Scaling.

Standardization also called Z-score Normalization.

Formula:

$$x_i = \frac{x - \bar{x}}{\sigma}$$

Mean centering & scaling by σ.



$\boxed{\text{Mean} = 0,\ SD = 1}$

eg, logistic Regression etc.

When to use Standardization ⇒

(1) K-Means, (2) KNN (K-Nearest Neighbour), (3) PCA (Principal Component Analysis)

(4) Artifical Neural Network, (5) Gradient Descent

→ used in majority of Neural Network.

Not used in: Decision Tree, Random Forest, Gradient Boost, Xgboost. etc.

(25) Lecture # 25 (video-25) → 24 min

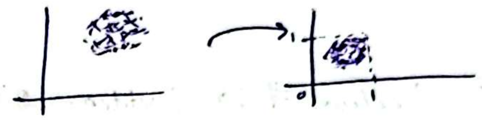
→ Normalization, In this the units of any numerical values is normalized eg, 10 kg, 100 g etc → convert this into standard unit and then proceed.

Types of Normalization:
- ↳ Min-Max Scale.
- ↳ Mean Normalization.
- ↳ Max absolute Scaling.
- ↳ Robust Scaling.

→ Min-Max Scaling →

Formula. $x_i' = \frac{x_i - X_{min}}{X_{max} - X_{min}}$

Distribution Must be Scaled between = [0, 1]
↓ Min ↓ Max.



→ Not good when there is outlier.

→ Mean Normalization →

$X_i' = \frac{x_1 - X_{mean}}{X_{max} - X_{min}}$ (Mean centering)

→ [-1, 1]

→ There is no specific library for that.

→ Required for centered data.

→ MaxAbsolute Scaling: (library sklearn MaxAbsScaler)

$X_i' = \frac{x_i}{|X_{max}|}$

Used in sparse data
↳ data having many zeros values.

→ Robust Scaling,

$x_i' = \frac{x_i - X_{median}}{IQR\ (75^{th}\ percentile - 25\ percentile)}$

Library in sklearn ⇒ Robust Scaler.

Used →
↳ Robust to outlier.

→ Majority Standardization is used.

→ Min-Max Scaler → images → $\boxed{CNN}$
↓ 0-255.

↓ if you know the min and max value of Data, use this.

(26) Lecture # 26 (video-26) → 20 min

→ Encoding Categorical Variable:

Data
- → Numerical Data
- ↳ Categorical Data
  - → Nominal Data → eg, State, sections etc
  - ↳ Ordinal Data → eg, order or relationship like reviews etc.

Label Encoding → when the output data should be in categorical data.

eg: classification etc.

Types of Encoding,
- → Ordinal Encoding → used with ordinal data (values have relation)
- ↳ One Hot Encoding → used with nomial data (the values of the column does not have relationship)

(27) Lecture # 27 (video- 27) → 30 min.

→ One-Hot Encoding (OHE).

↳ forms vector.

eg, Yellow, Red, Blue.

[1,0,0], [0,1,0], [0,0,1].

Dummy variable Trap ⇒

After OHE we remove one column probably first column.

Multicollinearity ⇒ faces you should not have or let mathematical relation b/w columns.

⇒ OHE using most frequency variable ⇒

what if there are many variables like suppose 40 brands, then we make some frequent values and make one column named others. eg, 10 brand frequency used, 30 brand lies in other column

(28) Lecture # 28 (video - 28) → 16 min

→ Column Transformer. → All pratical.

(29) Lecture # 29 (video -29) → 46 min

→ Piptelines.

↳ output of every step is input for next step.]- All pratical.

(30) Lecture # 30 (video - 30) → 32 min

↳ Feature Transformation

↳ Mathematical Transformer.

↳ Log Transform.

↳ Reciprocal Transform.

↳ Power Transform.

↳ Box - Cox.

↳ Yeo- Johnsm.

Sklearn library

- Function Transformer.
  - log Transform.
  - Reciprocal. Transform.
  - Power Transform.
- Power Transform.
  - Box - Cox.
  - Yeo. Johnsm.

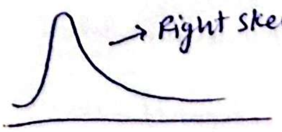
How to check whether the data is skew of or not?

sns.displot.

QQplot.

pd.skew()

Log Transform → used in right skewed data.



→ Right skewed.

Reciprocal Transform →

$1/x$ →

$sq(x)^2$ → left skewed data.

sqrt $\sqrt{x}$

→ left skewed.

(31) Lecture # 31 (video - 31) → 21 min

Power Transformer.
- → Box-Cox
- → Yeo-Johnson.

Box-Cox, $x_i^{(\lambda)} = \begin{cases} \dfrac{x_i^{\lambda} - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(x_i) & \text{if } \lambda = 0, \end{cases}$ Range (-5 to 5).

Restrictly Applicable on $\boxed{n > 0}$.

Yeo-Johnson:
Can apply on negative as well as positive values.

Box-Cox → library used → PowerTransformer(method='Box-Cox')

Yeo-Johnson → library used → PowerTransformer()
↳ Default value Yeo-Johnson.

For QQ plot used library → scipy.stats.

(32) Lecture # 32 (Video-32) → 39 min

→ Binning & Binarization (Numerical to Categorical Data).
(Discretization)

Binning

forming range
like
10, 1000, 200, 400, 500, 2 etc

0-10, 10-50, 50-500, 500-1000, 1000+
Bin, Bin, Bin, Bin, Bin.

Types of Binning,
- ↳ Unsupervised Binning.
  - Equal width (uniform) Binning.
  - Equal frequency (quantile) Binning.
  - KMeans Binning.
- ↳ Supervised Binning
  - Decision Tree Binning.
- ↳ Custom Binning.

→ (Equal width Binning)

Number of Bins = ? You will decide

Formula = $\dfrac{\text{max} - \text{min}}{\text{No. of Bins}}$

Benefits:
- ↳ Handling outlier
- ↳ No change in spread.

→ Equal Frequency / Quantile Binning (Default).

Divide data into percentile (Interval) Usually more used.

Benefit:
- ↳ Handling outlier.
- ↳ Make spread value uniform.

KMean Binning = makes cluster

↳ library → sklearn.KBinsDiscretizer()
- ~~bins(+)~~ bins = ?
- Strategy
  - uniform
  - quantile
  - KMeans
- Encode
  - ordinal
  - One-Hot Encoding.

Binarization → convert value in 0 or 1.
- ↳ used in specfic areas.
- ↳ library used sklearn.Binarizer()
  - Threshold: above it value One below it zero
  - copy → make ...

→ Custom / Domain Based Binning ⇒

like { [0 - 18] → kids
[. . . . }

No library
Pandas
↳ library used for it.

(33) Lecture # 33 (video - 33) → 12 min

↳ Mixed Data (Handling Mixed values)

like Cabin

Cabin
B5
C23
D41

in which B is class no. and 5 is room no. or seat no.

→ Two new column

B	5
C	23
D	41

Column value

```
Code: pd.to_numeric(df[ ],
      errors = 'coerce',
      downcast = 'integer').
```

(34) Lecture # 34 (video - 34) → 14 min

↳ Handling Data and Time Based column.

↳ Pratical work.

→ month, year, day, quater, semster, week, week-day, isweekend, etc.

(35) Lecture # 35 (video - 35) → 25 min

↳ missing value (Handling Missing value.

- Remove That column or row.
  - (CCA) Complete Case Analysis
  - (Do it only when
    - ① MCAR Missing completing at random
    - ② if 5% Less data is missing.)
- Impute (Fill)
  - → Univariate (Simple Imputer class)
    - → Numerical Data
      - → Mean/Median.
      - → Random value.
      - → End of Distribution value.
    - → categorical Data.
      - Mode.
      - Missing.
  - → Multi-variate
    - → KNN Imputer
    - → Iterative Imputer

(36) Lecture # 36 (video-36) → 31 min

↳ Complete Missing value (Numerical data)

Univariate Imputation → Numerical Data
- Mean/Median.
- Random value/Arbitatory
- End of Distribution
- Arbitrary value.

① Mean/Median Imputation → Replace with mean & median.
- mean: use when Normal Distribution
- median: use when skewness

↳ Disadvantage.
1) Distribution shape change
2) Outlier formation
3) Covariance/corr relation change.

When to use,
1) MCAR
2) Missing value should be less than 5%.

② Arbitrary Value Imputation → Replace value with missing value or any value.

Benefit → easy to apply.

Disadvantage, → PDF distort.

→ Variance changes → Covariance change etc.

③ End of Distribution Imputation

↳ Replace The null value with

Normal Data → Formula = (mean + 3δ)
OR
(mean − 3δ)

Skewed ~~Value~~ Data. → Formula:
↓
$Q_1 = 1.5 IQR$
$Q_3 = 1.5 IQR$
$IQR = Q_3 - Q_1$

(37) Lecture # 37 — (video-37) → 13 m

~~④ Random Sample Imputation~~

↳ Handling Categorical Missing Data

① Most-Frequent Value Imputation,
→ Replace data with most frequent value (mode)

Disadvantage → change data distribution

② Missing Category Imputation:
↳ Replace data with 'Missing' value. named as 'Missing'.

If data is 10% greater missing.

(38) Lecture # 38 — (video-38) → 38 min.

↳ Handling Missing values. (Both on Numerical & Categorical)

① Random Imputation : → Replace or fill by any random. by seeing the data.

② Missing Indicator → Replace with new value (Make T/F).
↳ Making the difference in the null or fill value.

③ Automatically select value for Imputation
↳ use gridsearch CV. for best Impulation technique.

(39) Lecture # 39 (video-39) → 25 min.

↳ Handling Missing values. (Multivariate Imputation)

① KNN Imputer ⇒ Replace with the closest neighbour of the null value
↳ Euclidean distance

Advantage
↳ More Accurate technique

Dis-advantage
↳ Complex Calculation.
↳ Production Deployment
↳ You have to deployment all dataset.

(40) Lecture # 40 — (video-40) — 18 min

↳ Iterative Imputer (MICE)

MICE → Multivariate Imputation by Chained Equations.

Used for → MAR → Missing at Random.
→ Accurate Result.
→ Disadvantage.
same in KNN.

Steps
① Replace with mean value. (Itr1)
② Replace again with nan. (Itr2)
③ Applying model (Any).
→ Now do same on all columns
④ Subtract Itr1 − Itr2 Then (make This To that The difference will be zero.)
By updating.

41) Lecture # 41 (video - 41) → 18 min

↳ What are outliers?

⇒ Effect/Impact of Outlier on:
1) LR, Logistic R.
2) Deep learning.
3) Adaboost etc.

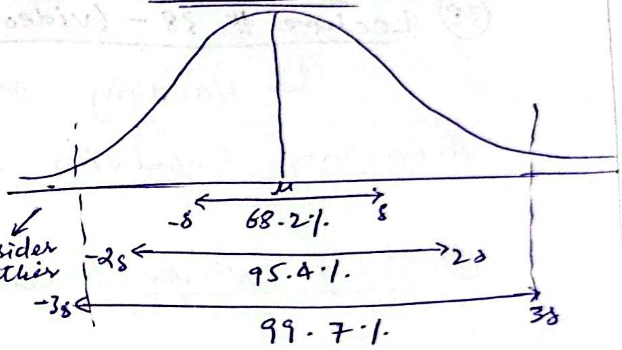
→ Treat Outlier
→ Trimming
→ Capping
→ Missing value.
→ Discretization (Bins)

Detect Outlier
→ Normal Distribution
→ Skewed Distribution
→ other Distribution

Techniques
→ Z-score tech
→ IQR Based Fitting.
→ Percentile
→ Winsorization.

42) Lecture # 42 (video-42) → 18 min

Outlier Detection & Removal (Z-score technique For Normal Distribution).



Formula used,

Outlier = $(\mu + 3\delta > $ Outlier $< \mu - 3\delta)$

Zscore:

$$x_i = \frac{x_i - \mu}{\delta}$$

Treatment of Outlier:

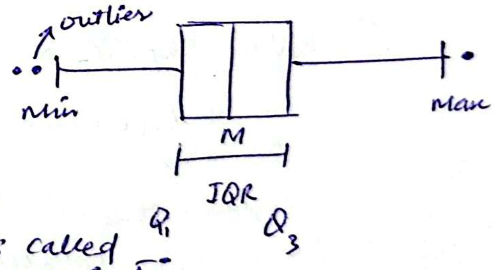
Trimming → Remove value.

Capping → Cap to the Max/Min value.

43) Lecture #43 (video-43) → 14 min

Outlier Detection & Removal (IQR Technique For Skewed Distribution).

44) Lecture #44 (video-44) → 16min



Outlier Detection using Percentile Method.

General value could be 99% or 1%, it totally depend on ~~the~~ you. In this capping is called winsorization.

45) Lecture # 45 (video - 45) → 12 min

↳ Feature Engineering → Feature Construction

→ Make New column by combining different column.

→ OR splitting data/column into different column.

46). Lecture # 46 (video - 46) → 16 min.

↳ Feature Selection (Curse of Dimensionality).

⇒ Curse of Dimensionality (COD)

feature → Dimensionality

├ Images, text data etc.

Sparsity → The datapoint moves farther from each other.

Solution → Dimensionality Reduction.

Feature Selection.
1) Forward selection.
2) Backward selection.

Feature Extration
1) PCA
2) LDA 3) TSNE

47) Lecture # 47 (video. 47) → 34 min.

↳ Feature Extraction ⟶ PCA (Unsupervised Leaning Technique).

↳ Higher Dimension to Lower Dimension data.

* Benefits of using PCA ⇒

1) Faster execution of algo.
2) Visualization.

How To select Feature,
→ Check the shadow/spread of Data.
→ Variance (to check spread).

48) Lecture # 48 (video-48) → 57 min.

→ PCA (Part-2) Continious: Mathematical computation in Detail.

↳ Variance, Covariance & Covariance Matrix, Eigan vectors, Eigan values etc.

↳ Magnitude change Direction remains same.

Step By Step PCA

1) Mean Centered → Data.
2) find Covariance matrix.
3) Find The Eigan vector & Eigan value

49) Lecture # 49 (video-49) → 43 min.

↳ Pratical work on MNIST Dataset.

For Visualization used Library ⇒ plotly.express, Pandas Profiler, IPWidgets etc..

→ optimium Number of PCA component → By variance.

50) Lecture # 50 (video-50) → 34 min.

↳ First Algorithm ⟶ Simple Regression (Linear Regression)

Simple LR | Multiple LR | Polynomial LR | Regularization.

(51) **Lecture 51 (video-51) → 53 min**

↳ Simple Linear Regression (code - Make class of simple LR)

→ Closed-Form Solution → ability to solve technique by mathematical Formula.
↳ OLS (ordinary least square).

→ Non-Closed-Form Solution → Gradient Descent.

→ LR Mathematical Function:
My notes of IDS + Prob etc.

(52) **Lecture 52 (video-52) → 43 min**

Metrices to check Regression Accuracy.

- MAE → Mean Absolute Error → $\boxed{MAE = \sum_{i=1}^{n} |y_i - \hat{y}_i| / n}$
- MSE → Mean Squared Error → $\boxed{MSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 / n}$
- RMSE → Root Mean Squared Error. → $\boxed{RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}}$
- R2 Score → Co-efficient of Determination. → $\boxed{R_2 = 1 - \frac{SSR}{SSM}}$
- Adjusted R2 score →

1) MAE
Benefit.
1) give same units.
2) Robust Outliers
Disadvantage.
1) Modulus Function graph is not differentiatable

2) MSE.
Benefit
1) Used as loss Function.
Disadvantage.
1) Not same unit.
2) Not Robust to outliers. (Huge impact).

3) RSME.
Disadvantage
↳ Robust outliers.

R2-Score
Calculating The difference b/w mean and Regression line.

R2 Score:
~~Benefit~~
R_2 score should close to 1.
Disadvantage
Sometimes R_2 score increase with increase in no. of columns.

Adjusted R2 Score
$$\text{Adjusted } R_2 \text{ Score} = 1 - \left[\frac{(1-R_2)(n-1)}{(n-1-k)}\right]$$
n → no. of rows.
k → independent.

(53) **Lecture 53 (video-53) → 21 min**

↳ Multiple Linear Regression (Part-1)

(54) **Lecture # 54 (video-54) → 48 min.**

↳ Multiple Linear Regression (Part-2).

OLS → LinearRegression.
Gradient descent → SGDRegression

(55) **Lecture # 55 (video-55) → 16 min.**

↳ Multiple Linear Regression (Part-3) (code — Making class of MLR)

(

(56) Lecture 56 (video-56) → 1 hour 57 min

↳ Gradient Descent Complete.

↳ Types

↳ Batch GD.

↳ SGD (stohastic)

↳ MBGD. (mini Batch)

Batch Gradient Descent (2D)

→ Make learning rate according to Data.

of b,

$\boxed{\text{Slope} = -2\sum(y_i - mx_i - b)}$

~~$b_{new} = b_{old} -$~~

of m,

$\boxed{\text{Slope} = -2\sum(y_i - mx_i - b)x_i}$

→ Convex Function.

↳ contain only one global minima.

→ Loss Function Problem.

(57) Lecture 57 (video-57) → 1 hour 4 min

↳ Gradient Descent → Type I (Batch Gradient Descent)

↳ above video is related to this.

* Batch GD (3D): or (N-D)

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \cdots \beta_n x_n.$

Random initialization,

1) Generally: $\beta_0 = 0,\ \beta_1, \beta_2 = 1.$

2) epoch = 100, lr = 0.1.

$\beta_0 = \beta_0 - \eta \times \text{slope}.$

learning rate

$\beta_1 = \beta_1 - \eta \times \text{slope}.$

$\beta_2 = \beta_2 - \eta\ \text{slope}.$

Loss Function:

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

General Formula of loss Function:

$$\boxed{\frac{\partial L}{\partial \beta_m} = \frac{-2}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)X_{im}}$$

np.sum. mean

(58) Lecture 58 (video-58) → 50 min.

↳ Stochastic GD → gives faster conversion.

↳ Almost same minor changes but it is used for Big data

↳ Converge faster than other.

When to use

↳ Big data → sgd.

↳ Non-Convex Function.

→ Vectorization, Learning schedule.

(59) Lecture 59 (video-59) → 22 min.

↳ Mini Batch GD.

(60) Lecture 60 (video-60) → 27 min.

↳ Polynomial Linear Regression.

→ Hyper-parameter,

CamScanner

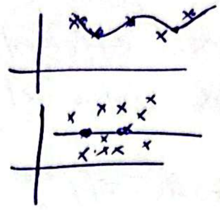
+ Lecture # 61 (video-61) → 8 min.

↳ Bias- Variance Trade-off.

↳ The inability of ML model to recapture the relationship of dataset.

→ Low Bias → High Variance

→ High Bias → Low Variance



Overfitting, give excellent result on dataset but is not good on test data.

Underfitting, give not good result on traing dataset.

Three types:

1) Regularization 2) Bagging 3) Boosting.

+ Lecture # 62 (video-62) – 18 min

↳ Regularization (Ridge Regularization) → Part 1.

Types: 1) Ridge (L2)

2) Lasso (L1)

3) Elastic Net. ()

$$L = \underbrace{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}_{\text{To minize This}} + \underset{\substack{\downarrow \\ \text{hyper-parameter} \\ \text{(can be tune it)}}}{\lambda} (m^2)$$

+ Lecture # 63 (video-63) → 43 min

↳ Ridge Regression (Part-2)

→ Making our own class. of Ridge. (OLS) Method

↳ using library (Ridgor)

$$\boxed{m = \frac{\sum_{i=1}^{n} (Y_i - \bar{Y}_i)(X_i - \bar{X}_i)}{\sum_{i=1}^{n} (X - \bar{X})^2 + \lambda}}$$

$$\boxed{b = \bar{Y} - m\bar{X}}$$

This parameter is added in ridge regression. If its value is 0, it would be simple LR.

In N-Darray:

Loss Function ⇒ $\boxed{L = (XW - Y)^T (XW - Y) + \lambda W^T W.}$

+ Lecture # 64 (video-64) → 19 min

↳ Ridge Regression (Part-3) → Applying Gradient Descent.

→ Gradient Descent (~~SD~~ SGD Regressor)

* Lecture # 65 (video- 65) → 30 min.

↳ (5 key Points of Ridge Regression)

1) Co-efficient get affected by Ridge (How)?

↳ The co-efficient always shrinks to zero but never be zero. (very close to zero).

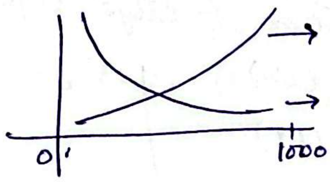
2) Higher values are affected more ?

↳ Higher value of coeff- → higher value to shrink.

3) Bias Variance Trade-off ?

↳ if $\lambda\downarrow$ Bias$\downarrow$ overfit Variance$\uparrow$.

↳ if $\lambda\uparrow$ Bias$\uparrow$ underfit Variance$\downarrow$.



→ Bias (small at low value, high at big value).

→ Variance (Big at low value, small at high value).

4) Impact on Loss Function?

↳ $\lambda\uparrow$ coeff $\downarrow$

5) Why called "Ridge"? → Hard constriant Rigde constraint

Tip: Always use Ridge value >= 2 inputs.

* Lecture # 66 (video- 66) → 29 min

↳ Lasso Regression. (Notes) → on laptop code

↳ It will do feature Selection.

* Lecture # 67 (video-67) → 25 min

↳ Why lasso make more sparsity.

Lasso Makes 3 cases

1) If $m>0$

$$m = \frac{\Sigma (y_i - \bar{y})(x_i - \bar{x}) - \lambda}{\Sigma (x_i - \bar{x})^2}$$

If $m=0$

$$m = \frac{\Sigma (y_i - \bar{y})(x_i - \bar{x})}{\Sigma (x_i - \bar{x})^2}$$

for $m<0$

$$m = \frac{\Sigma (y_i - \bar{y})(x_i - \bar{x}) + \lambda}{\Sigma (x_i - \bar{x})^2}$$

In Lasso → The λ is in numerator.

In Ridge → The λ is in denominator.

Lecture # 68 (video - 68) → 12 min

↳ ElasticNet Regression.

(Combination of Lasso & Ridge).

Lecture # 69 (video - 69) → 47 min

↳ Logistic Regression (Part-1) → Perceptron tricks.

How to label Regions? ↳ $AX_1 + BX_2 + C = 0$

C ↳ Y-intercept.

↳ By check it on desmos website.

Transformation:

↳ C↑ → completely change line.

↳ A↑ → etc.

→ If Negative point is in positive region Then substract with line equation.

→ If positive point is in negative region Then add with line equation.

Lecture # 70 (video-70) → 18 min

↳ LR (Part-2) (Notebook coding)

→ Perceptron Trick.

Lecture # 71 (video-71) → 40 min

↳ LR (Part-3) Sigmoid function.

$$\underset{\text{new}}{W_n} = \underset{\text{old}}{W_o} + \underset{\text{learning rate}}{\eta}(Y_i - \hat{Y}_i) X_i$$

Lecture # 72 (video-72) → 20 min

↳ LR (Part-4) Loss Function. (Maximum Likelihood).

→ Cross Entropy → The sumission of -ve logs.

↳ Minimum value → Best Model (If using cross-entropy)

$$\text{Loss Function} = \Sigma - Y_i \log(\hat{Y}_i) - (1 - Y_i) \log(1 - \hat{Y}_i)$$

$$\boxed{\left.\begin{array}{l}\text{Log-loss error}\\ \text{Binary Cross Entropy}\end{array}\right] - \frac{1}{n} \sum_{i=1}^{n} Y_i \log(\hat{Y}_i) + (1 - Y_i) \log(1 - \hat{Y}_i)}$$

Lecture # 73 (video-73) → 6 min.

↳ Derivation of Sigmoid Function.

$\sigma = 1 / 1 + exp(-z)$.

Sigmoid Function

Derivation ⇒ $\sigma(x) [1 - \sigma(x)]$

$$\boxed{\sigma'(x) = \sigma(x) [1 - \sigma(x)]}$$

↳→

Lecture # 74 – (video-74) → 37 min

↳ LR → (Part-5) → Applying on Gradient Descent.

$$\boxed{\text{Loss Function in form of Matrix} = -\frac{1}{m}\left[Y \log \hat{Y} + (1-Y)\log(1-\hat{Y})\right]}$$

where $\hat{Y} = \sigma(XW)$.

Gradient Descent for LR → $\boxed{W = W + \eta \frac{1}{m}(Y-\hat{Y})X}$

Lecture # 75 - (video-75) → 35 min.

↳ Classification Matrix.

→ Accuracy. (How Much Accuracy score should be?)

→ Confusion Matrix.

↳ It depends upon the model or problem on which you work.

		Predicted Value 1	0
Actual Value	1	True Positive	False Negative
	0	False Positive	True Negative

$$\boxed{\text{Accuracy Score} = \frac{TP + TN}{TP+TN+FP+FN}}$$

False Positive (FP) → Type I Error.

False Negative (FN) → Type II Error.

lecture # 76 . (video-76) → 42 min.

↳ Precision Recall & F1 Score.

Precision ⇒ ~~Pos~~ Proportion of predicted Positive is Truely positive?

Recall ⇒ Proportion of Actual Positive is correctly classified.

F1 score ⇒ ~~(Precision + Recall)/2~~ F1 score = $\frac{2PR}{P+R}$ P = Precision, R = Recall.

Macro-Precision ⇒ Get the combined precision. (Balanced classes).

Weighted Precision ⇒ Also multiply with weights (with unbalanced class).

Lecture # 77 (video-77) → 39 min

↳ Softmax Regression (Multi-nomial LR).

Softmax Function,

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Loss Function in Softmax ⇒

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} Y_k^{(i)} \log(\hat{Y}_k^{(i)})$$

Lecture # 78 (video-78) → 10 min

↳ Polynomial LR → Polynomial Features by increasing degree like degree = 2 ↑

Used in Non-linear dataset but RF & DT could result better.

* Lecture # 79 (video-79) - 13 min

↳ Logistic Regression(LR) Hyperparameters.

Sklearn libraries:

Parameters of LR:

penalty, dual, etc. Use Bookmark for it + Notes (link under video)

* Lecture # 80 (video-80) → 1 hour

↳ Decision Tree. (Geometric Intuition).

Game Based on DT → akinator

Entropy: Measure of disorder or measure of purity/impurity

For 2 class the min entropy = 0 & Max Entropy = 1.

$\log_2$. Those having less Peak have high Entropy.

leafnode → zero.

Gini Impurity: Measure the impurity in nodes.

Entropy: $-P_y \log_2(P_y) - P_N \log_2(P_N)$ → For 2 class.

Gini Impurity: $1 - (P_y^2 + P_N^2)$ → For large dataset (Fast).

* Lecture # 81 (video-81) → 28 min.

↳ Decision Tree Hyperparameters.

Hyper-parameter

↳ Depth of Tree (max-depth).

Hyperparameter → criterion, splitter, max depth, Min Sample split, min sample leaf etc.

* Lecture # 82 (video-82) → 35 min

↳ Regression Trees.

→ Can done hyper tuning by GridSearchCV or RandomCV.

* Lecture # 83 (video-83) → 19 min

↳ Library dtreeviz (used for tree).

→ Must use in case of Decision Tree or Random Forest etc.

* Lecture # 84 (video-84) → 38 min.

↳ Introduction to Ensemble learning.

→ Wisdom of The Crowd.

Ensemble learning ⇒ where multiple models combine together to form or predict

Types of Ensemble learning ⇒

① Voting Ensemble ⟶ Diff models (combines)
② Bagging ⟶ Random Forest
③ Boosting ⟶ Adaboosting.
⟶ Gradient Boosting.
↳ XgBoost.
④ Stacking.

→ Bagging stands for Boostrapped Aggregation.

Benefits:

→ Improvement in performance.

→ Bias & Variance (low)

→ Robustness.

Lecture # 85 (video-85) → 17 min
↳ Voting Ensemble

Lecture # 86 (video-86) → 24 min
↳ Voting Ensemble → Classifier/Classification.

→ Hyper-parameter → Voting type, Estimators etc.

Lecture # 87 (video-87) → 11 min.
↳ Voting Ensemble → Regression → Part-③.

Lecture # 88 (video-88) → 32 min.
↳ Bagging → Introduction.
(100 of models combine together)
Note: There should be only one model used.

Bagging Types.
→ Pasting (give good result)
→ Random Subspace.
→ Random Patches.

Lecture # 89 (video-89) → 23 min
↳ Bagging Ensemble → Part (2)

Bagging Classifier Hyperparameter:

↳ Base estimator, no. of estimator, Max sample, Boostrap samples,
Max Features etc.

$\boxed{\text{OOB Score}}$ etc ⟷

* **Lecture # 90 (video-90) → 11 min.**

↳ Bagging Ensemble → (Part - 3)

→ Bagging Regressor

* **Lecture # 91 (video - 91) → 34 min.**

↳ Random Forest.

└→ group of trees.

↓

Bagging → Boostrapped Aggregation.

Boostrapped ↓ Multiple models

Aggregation ↓ Majority of Answer wins.

* **Lecture # 92 (video - 92) → 13 min**

↳ Random Forest works well why?

↳ Bias Variance trade-off

Best Machine Learning Algo.

Low Bias
Low ~~High~~ Bias.

LB HV	HB LV	LB LV.
→ Fully Grown DT.	→ LR	→ Random Forest
→ SVM		
→ KNN		

→ Bias & Variance have inverse relation.

* **Lecture # 93 (video-93) → 13 min**

↳ Bagging Vs Random Forest.

Differences:

Bagging	Random forest
→ Can use different models. (Best is DT, KNN, SVM) python parameters: base_~~pa~~ estimator.	→ Only use $\boxed{DT}$. python parameter No base_estimator.
→ Tree-level column sampling.	→ Randomness. ↳ Node level column sampling.

* **Lecture # 94 (video- 94) → 16 min**

↳ Random Forest Hyperparameters.

→ Num_estimator, Max_feature, boostrap, max_sample, criterion, max_depth _etc_

- Num_estimator ↓ How many DT we made.
- Max_feature ↓ No. of column.
- boostrap ↓ random ness or not
- max_sample ↓ No. of rows.

* **Lecture # 95 (video - 95) → 12 min**

↳ Hyperparameter tunning of Random Forest.

→ GridSearch CV or RandomSearchCV is best used for hyper-parameter tunning.

* **Lecture # 96 (video - 96) → 7 min**

↳ OOB Evalution → Out of Bag evalution.

→ Almost 37 % of rows remains hidden in OOB. sample

↳ can use as validation set for checking accuracy.

* **Lecture # 97 (video - 97) → 28 min.**

↳ Feature Importance of Random Forest.

→ Importance of column in predicting the output.

* **Calculating Feature Importance (DT)**

↳ Node Importance

$$X_i = \frac{N_t}{N}\left[\text{impurity} - \left(\frac{N_{t\_r}}{N_t} \times \text{right\_impurity}\right) - \left(\frac{N_{t\_l}}{N_t} \times \text{left impurity}\right)\right]$$

5, 5, 0.48, 4, 5, 0.375, 1, 5, 0.0

X_i = 0.18

N_t → samples

N →

gini → impurity

X[1] <= -0.89
gini = 0.48
samples = 5
value = [3,2]

gini = 0.0
samples = 1
value = [0,1]

X[0] <= 1.01
gini = 0.375
sample = 4
value = [3,1]

0^{th} ~~san~~ column $= \frac{X_i}{X+Y} \longrightarrow \frac{0.18}{0.18 + 0.37}$

Same do for y.

* **Lecture # 98 (video- 98) → 18 min.**

↳ Adaboost Classifier

→ Weak learner → Just over 50 % is called weak learner

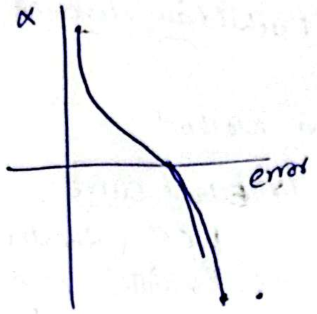
→ Decision stumps → is type of weak learner.

→ Stage-wise (additive Method)

formula: $h(x) = \text{sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x))$

* Lecture # 99 (video-99) → 20 min.

↳ AdaBoost → step by step Explanation



Formula = $\alpha = \frac{1}{2} \ln\left(\frac{1 - error}{error}\right)$

→ Upscaling → by giving more weightage To mis-classified points.

For mis-classified:

$new\_wt = curr\_wt \times e^{\alpha_1}$

For classified:

$new\_wt = curr\_wt \times e^{-\alpha_1}$

* Lecture # 100 (video-100) → 17 min.

↳ AdaBoost_Demo → Pratical Notebook.

Algorithm.

* Lecture # 101 (video-101) → 12 min

↳ Adaboost Hyperparameter.

→ Learningrate, n_estimator etc. hyper-parameters.

* Lecture # 102 (video-102) → 7 min.

↳ Bagging Vs Boosting.

Bagging	Boosting.
→ Model used (LBHV) ↳ Fully grown DT. ↳	→ Model used: (HBLV) ↳ Shallow DT. ↳ Decision stump.
→ Sequential Vs Parallel Bagging uses Parallel.	→ Boosting uses sequentially.
→ weightage: equally weightage.	→ weightage. weight assigned, not equally, best model provide better weightage.

* Lecture # 103 (video-103) → 24 min

↳ K-Means Clustering Algorithm.

Steps.
→ Decide n clusters.
↓
→ Init Centroids.
↓
→ Assign cluster.
↓
→ Move centroids
↓
Finish.

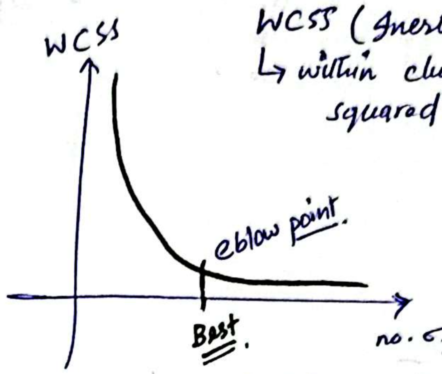
→ Euchildian distance.

→ Elbow Method

↳ Elbow Curve

WCSS (Inertia)

↳ within cluster sum of squared.



* Lecture # 104 (video-104) → 10 min

↳ K-Means clustering Algorithm code.

→ Practical Notebook work.

* Lecture # 105 (video-105) → 34 min

↳ K-Mean clustering Algo from Scratch.

Eucheldian Distance. $\sqrt{(x_2-x_1)^2+(y_2-y_1)^2} \ldots$

↳ np.sqrt(np.dot(a-b, a-b))

* Lecture # 106 (video-106) → 32 min

↳ Gradient Boosting.

Loss Function ⇒ Acutual − Predicted.
(pseudo-residual)

Pred = $m_1 + m_2 \times lr$

Pred1, Pred2

Model 1 should be means

Model 2 predicts The residual

learning rate should be added after 1st model.

AdaBoost Vs Gradient Boost.

AdaBoost	Gradient Boost
Decision stumps	Max leaf nodes
Weightage given ↓ Different for every model.	learning rate given ↓ Constant for every model.

* Lecture # 107 (video - 107) → 57 min.

↳ Gradient Boosting Algorithm (Mathematical Intuition).

Regression Problem.
↓
Loss Function: Least square.

→ Additive Modelling → Add small function and make sum of it.

① We should have training data, Loss function (Differentiation), no. of iteration.

Should be in loop → ② Initialize: $f_0(x) = \arg\min_{\gamma} \frac{1}{2} \sum_{i=1}^{n} (y_i - \gamma)^2$.

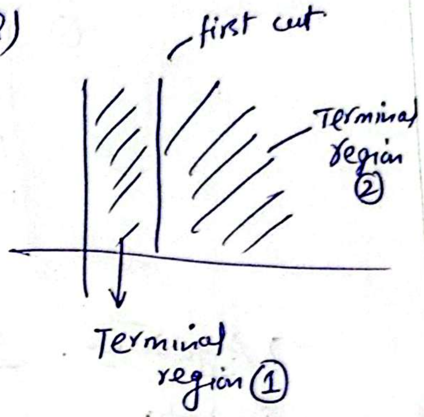
(a) Find mean of output. (leaf function).

(b) Calculate Pseudo-residual (Residual)

(c) Predict 2nd model. (Regression Tree / Decision Tree)

(d) Update DT output.

③ Output of DT.



* Lecture # 108 (video-108) → 1 hour 4 min.

↳ Gradient Boosting.

Classification Problem
↓
Loss Function: log loss.

① 1st model → $\log\left(\frac{\#1}{\#0}\right)$ — log (odds) — no. of #1 / no. of #0

Like if we have output
1's → 6.
0's → 5 Then $\log\left(\frac{6}{5}\right)$

② 2nd model Prediction Formula:

$$\frac{\sum \text{Residual}}{\sum(\text{Previous Probability} * (1 - \text{Previous Prob}))}$$

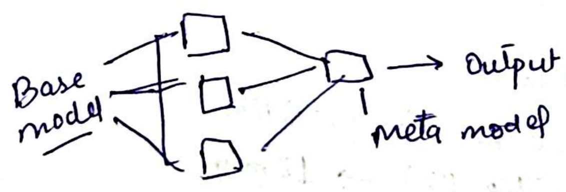
Probability:

$$P = \frac{1}{1 + e^{-\log(\text{odds})}}.$$

* Lecture # 109 (video-109) → 36 min.

↳ Stacking and Blending Ensemble.

→ meta model



Difference b/w bagging & Boosting

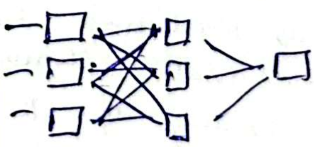
→ Different Base model

→ Training the output on new model

→ Hold-out Method
↳ Blending
↳ Divide the train data into Two parts.
↳ one used for training
↳ one used for validation

→ K-fold Method
↳ Stacking
→ Divide the train data into K fold (cut).
↳ (K−1) used for training
↳ one used for validation

Multi-layer stacking (Blending).



Note: Each model is trained on the no. of K-fold

* Lecture # 110 (video - 110) → 38 min.

↳ Agglomerative Hierarchical Clustering.

Clustering Methods

→ Hierarchical clustering.

↳ Density Based clustering (DB clustering).

→ Hierarchical Clustering Types

① Agglomerative clustering → Consider everypoint cluster itself.
→ Then consider two close point cluster.
→ Then more close. and so on...
→ Dendogram (Tree like structure)
→ Types.
→ Single Link.
→ Complete Link (Max)
→ Group Average
↳ Ward ↘ Default value.

② Divisive clustering.
→ works opposite to agglomerative.
→ Start with one cluster and continue to everypoint cluster itself.

→ Algorithm.

~~① Intro~~

→ Dendogram
↳ To find the no. of cluster.

Hypesparameter: n_cluster, affinity, linkage, distance_threshold etc.

→ No used for Bigger Dataset.

* Lecture # 111 (video - 111) → 52 min.

↳ K- Nearest Neighbour (KNN)

How to select k in KNN?

↳ Heuristic (Jugaad) → Experimented by others scientist.

↳ Experimentation → could be better to select this.
↳ cross - validation

Decision Surface → Tool → used classification problem

Limitations of KNN.
1) For large dataset (Not applicable)
2) High Dimensional data. (More columns).
3) Number of outliers.
4) Non-homogenous scales. (Distance b/w two features large).
⑤ Imbalanced dataset
⑥ Inference and not for prediction.

* Lecture # 112 (video-112) → 18 min.

↳ Assumptions of Linear Regression.

① Linear Relationship between Input & Output.
② No Multi-collinearity.
③ Normality of Residual.
④ Homoscedasticity.
⑤ No Autocorrelation of Errors.

* Lecture # 113 (video - 113) → 12 min

↳ Support Vector Machine (SVM).

↳ Geometric Intuition.

→ Margin maximizing hyperplane.

→ Kernels → To handle Non-Linear or robust outliers.

* Lecture # 114 (video - 114) → 35 min

↳ Mathematics of Support Vector Machine (SVM)

$\boxed{\vec{w} \cdot \vec{u} + b \geq 0}$ $\boxed{W^T x + b = 0}$

$d = \frac{2}{\|w\|}$

→ Hard-Margin SVM ⇒ contains separate data (clearly)

* Lecture # 115 (video - 115) → 15 min

↳ Soft Margin SVM.

$$\underset{(w^*, b^*)}{\operatorname{argmin}} \frac{\|w\|}{2} + \underbrace{c}_{\text{Hyper-parameter}} \sum_{i=1}^{n} \xi_i$$

zeta

Hinge Loss.

$\xi_i = 0$ (for correctly classified point)

* Lecture # 116 (video - 116) → 14 min

↳ Kernel Trick in SVM

→ Radial Basis Function.

* Lecture # 117 (video - 117) → 6 min

↳ Kernel Trick in SVM → Geometric Intuition.

* Lecture # 118 (video - 118) → 9 min

↳ Naive ~~Based~~ Bayes Classifier (Part - 1)

→ Conditional Probability.

* Lecture # 119 (video - 119) → 8 min

↳ Naive Bayes Classifier (Part - 2)

→ Independent Events in Probability / Mutually Exclusive

* Lecture # 120 (video - 120) → 1 min

↳ Naive Bayes Classifier (Part - 3)

→ Mutually Exclusive Event.

* Lecture # 121 (video - 121) → 5 min

↳ Naive Bayes Classifier (Part - 4)

→ Bayes Theorem in Probability.

* Lecture # 122 (video-122) → 9 min.

↳ Naive Bayes Classifier (Part-5)

→ Problem Based upon Bayes theorem.

* Lecture # 123 (video-123) → 15 min

↳ Naive Bayes Classifier (Part-6)

→ Intuition. (Naive Baye Theorem)

* Lecture # 124 (video-124) → 20 min

↳ Naive Bayes Classifier (Part-7)

→ Mathematics Behind Naive Bayes Algorithm.

→ Conditional Independence.

* Lecture # 125 (video-125) → 17 min.

↳ Naive Bayes Classifier (Part-8)

→ Simple Code Example.

* Lecture # 126 (video-126) → 9 min.

↳ Naive Bayes Classifier (Part-9)

→ Handling The Numerical Data for Naive Bayes.

↳ Guassian Naive Bayes

↳ Binomial, Multi-nomial Bayes.

↳ Possion Naive Bayes

* Lecture # 127 (video-127) → 1 hours 19 min.

↳ XgBoost.

→ Introduction → was developed in 2014.

XgBoost → use Gradient Boosting → have flexibility (Loss Function).

- → Performance.
- → Robust
- → Kaggle.

Was open-source in $\boxed{2016}$.

- Regression.
- Classification.
- Ranking.
- Custom Function etc

→ Optimization
- Parallel Processing.
- Optimized Data Structure.
- Cache Awareness.
- Out of core computing.
- Distributed Computing.
- GPU Support

Performance
- Regularized Learning objective.
- Handling Missing values.
- Sparsity effect.
- Efficient Split Finding.
  - Weighted Quantile Sketch
  - Approx Tree Learning
- Tree Pruning.

* Lecture # 128 (video-128) → 48 min.

↳ XgBoost (Part-2)

→ Simple Example Code.

Distributed Computing

↳ pyspark/spark used.

Main Different is:

XgBoost uses ~~Decesi~~ Decision tree which are different from normal gradient Boosting but the process will be same (Means not the Vanilla DT {gini, entropy}. use nhi krty.) ↓ estimators

Here we use:

$$\text{Similarity Score (SS)} = \frac{(\text{sum of residuals})^2}{\#\text{no. of residual} + \lambda}$$

→ Exact Greedy Search Algorithm.

* Lecture # 129 – (video-129) → 40 min.

↳ XgBoost for Classification. (Part-3)

→ XgBoost works on Classification.

* Lecture # 130 – (video-130) → 1 hours 57 min

Similarity Score different → in XgBoost

Regression

$$\left[\text{Similarity Score} = \frac{(\Sigma \text{Residuals})^2}{N+\lambda}\right]$$

$$\left[\text{Output Value} = \frac{\Sigma \text{Residuals}}{N+\lambda}\right]$$

Classification

$$\left[\text{Similarity Score} = \frac{(\Sigma \text{Residuals})^2}{\sum^{n}(P(1-P))+\lambda}\right]$$

$$\left[\text{Output Value} = \frac{\Sigma \text{Residuals}}{\sum^{n}(P(1+P))+\lambda}\right]$$

$(Y_i, \hat{Y}_i) \rightarrow$ relation

↳ Loss Function.

↓

Minimize

Objective Function:

~~Tytor~~ S. Taylor Series.

$$\mathcal{L} = \sum_{i=1}^{n} L(Y_i, \hat{Y}_i) + \Omega(f_k).$$

* Lecture # 131 – (video-131) → 35 min.

↳ DBSCAN Clustering Algorithms (Density Based Clustering)

⇒ Hyper-parameters ⇒ epsilon, minpts etc.

* Lecture # 132 – (video-132) → 58 min.

⇒ Imbalanced Data – in ML.

⟷