Subject : Machine Learning

Date : 10 9

Phase 0: Prerequisits

Mathematics for understanding Models.

Debug effectively. Train as a real scientist

Scikit Learn

Mean: The average of a set of numbers

Median: The middle value in a sorted set of numbers.

Mode: The most frequent value in a set of numbers.

Percentages, Ratios, Basic algebra, Plotting x vs y, Probability, vectors, Standard deviation.

Subject : _____ Date : __ __ __

Mean = $(x_1 + x_2 + \dots x_n) / n$

median = middle value (if odd count),

if even (middle 2 average)

mode = most repeated value.

If our dataset is unbalanced, our model can give us wrong prediction.

100 students

95 boys — 5 girls

(This is an example of unbalanced dataset)

100 students

50 boys — 50 girls

(1 balanced dataset).

Subject : _____ Date : ~~/~~/~~/~~

Compare actual value vs Predicted value.

Students	Actual Mark	Predicted Mark	Error
A	60	80	20
B	~~80~~90	85	-5

These maths are also helpful for analyzing error

5 5 10 50 1000

Here 1000 is outlayer. Because 1000 is way to much higher compared to other values.

Subject: Date: / /

Too much low values can be also outlayer. In this example it can be -100.

Cleaning outlayers helps the model to work efficiently.

Percentages and Ratio: Expressing numbers as parts of 100 is Percentage.

Compare two or more quantities in ratio.

Accuracy = (connected prediction / Total pred) X 100

Precision = (True positives / Total predictive positives)

Recall = True positives / Total actual positives)

Subject : _____ Date : 12 / 9

Basic algebra

$y = mx + b$

Ex: 500 daily income

1000 Joining bonus (Fixed)

0 - 1000 1 - 1500 2 - 2000 3 - 2500

y = Final salary predict, (output/result)

x = days worked, (input)

m = 500 (day, rate of change), slope

b = 1000 bonus. (starting point)

$y = 500 \times x + 1000$

4 days

$y = 500 \times 4 + 1000 = 3000$

Subject : _____ Date : __ __ __

Plotting x vs y

Hourse of studies	Marks
1	40
2	50
3	60
4	75
5	90

x = independent variable

y = dependent variable

Pattern here = More hours = More Marks

We have to find a pattern

Subject : _____ Date : ☐☐☐

Vectors :

g_3 R_1 R_2 R_3

g_2

g_1

[2.4]

Vector means a list of numbers

that describes details about a thing

ht. wt

(160 / 60) = height and weight

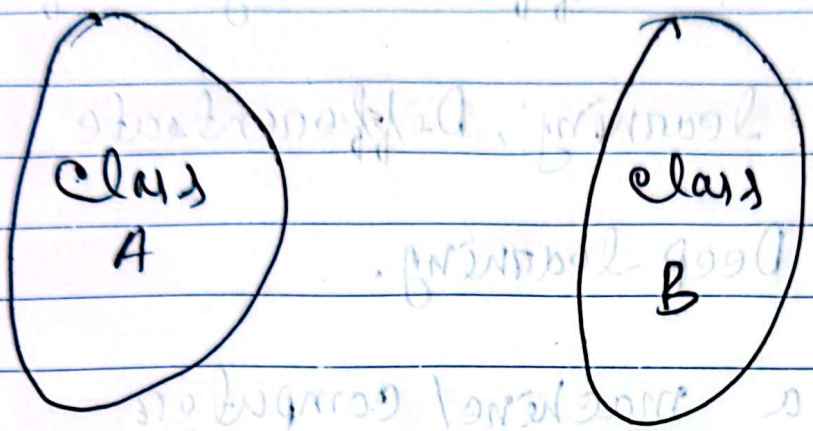
A single data can be stored as

Vectors

A whole dataset = Bunch of vectors.

Subject: Date:

Standard Deviation:



70, 72, 68 30, 70, 100

Close data spread data

Standard deviation measures whether data is closed to each other or spread.

To analyze dataset we use standard deviation. consistency data prep. data normalization and cleaning outlayer, We use .SD.

Subject : Date :

Phase 01 - Foundation of ML

What is ML ? Different types of ML, Scikit-learning, Differentiate AI, ML and Deep learning.

ML = When a machine/ computer learns from past data, pattern to make prediction or decision.

Machine learns from historical / past data.

AI is layer 1 or outer layer.

AI is a kind of technique that helps a machine to act smart.

Deep learning :- Deep learning try to mimic a human brain using neuron.

Subject : _____ Date : -- -- --

Self driving is an example of DL.

AI → Entire school - with rules

ML → A class → without rules

DL → A robot with a brain.

Scikit learning helps to predict.

Types of ML

i) Supervised learning: Ideal for tasks with labeled data where the desired output is known.

ii) Unsupervised learning: Suitable for identifying patterns in unlabelled data without explicit guidance.

HEART

Subject : _____ Date : __ __ __

iii) Reinforcement Learning: Best for training agents to make decisions through interaction with an environment.

Supervised learning: Price prediction Number, Age prediction.

Tools: Scikit learning: Leaneur and logistic regression, KNN, Decision Tree. I/O + O/P

Unsupervised learning: Clustering based on pattern. Only input. Machine will try to find pattern/ group:

K-means, PCA, DBScan.

We can: Marketing, Fraud detection

Subject : Date : -- -- ----

Reinforcement Learning: Machine is learning over the time. Try/error.

Ex: Gamer_Tesla, Robotics.

Action + Reward.

Algo: Dynamic Programming, Monte Carlo, ~~Temporda~~ Temporal Difference.

Core ML concepts

Input (x), output (y) X is always capital, y is small.

Model: A formula that helps to Predict.

Training: Understand the pattern.

Testing: Giving new data for ~~expect~~ Prediction.

Prediction: Final answer.

HEART

Subject : _____ Date : __ __ __

Scikit learn gives us ready made ML models. It is a python library.

1) Most widely used library

Phase 2 : Data PrePrcessing

(Preparing Data)

Raw Data

↓

Handle missing data

↓

Encoding Categorical values

↓

Feature Scaling

↓

Split Data

↓

Prepared data

HEART

Subject: •

Date: / /

Handle missing data : using Pandas.

Encoding categorical values ; categories to numbers. Text - Numerical data.

Feature scaling : Standard scaler or ~~Stand~~ min, max scaler.

Split data : For achieving fairness.

Methods for cleaning data (Missing value)

i) Drop Na

ii) Fill Na(value)

iii) isnull. sum

Subject : _____ Date : __ __ __

A simple program by Pandas

```
import Pandas as Pd
data = {      # some data
              }
df = Pd.Dataframe(data)
Print("Original Dataframe:", df)
Print(df.isnull().sum())
# it will provide us how many data
is missing.
df_drop = df.dropna()
Print(df_drop)
# drop rows with any null values.
```

Subject: _____ Date: __/__/____

For filling:

df['Age'].fillna(df['Age'].mean(), inplace
=True)

It will fill missing value.

Always try to fill data using mean
or median when working with Numerical
data.

For categorized data, use mode

Do not drop data blindly, try to fill.

Subject : . . .

Date : 12 | 9 | 2

Encoding Categorical Values

i) Label encoding.

ii) One-hot encoding.

Encoding values :

i) Non numerical values (strings, enums) into numeric features the model can use. Most ML algorithms expect numeric, continuous inputs. Raw string doesn't works.

Label encoding

Gender	encoded
Male	0
Female	1

Replace each category with a number.

HEART

Subject: _____ Date: __ __ __

apply: Gender Column

Lets apply Label encoding:

```
from sklearn.PreProcessing import LabelEncoder
import pandas as pd
df = pd.read_csv("sample.csv")
df_label = df.copy() #making a copy
le=LabelEncoder()
df_label = LabelEncoder()
df_label['Gender_Encoded']=le.fit_transform
(df_label['Gender'])
Print('\nLabel Encoded)
Print(df_label[['Name','Gender','Gender
_Encoded']])
```

Subject : _____ Date : __ / __ / __

One hot encoding turns a categorical feature into a set of binary indicator feature.

i) Original feature city with values [Dhaka, Noakhali, Camilla, Tangail]

ii) One-hot features: City = Dhaka, city = Noakhali, City = Tangail.

iii) A row with city = Noakhali becomes [0, 1, 0]

```
df_encoded = Pd.get_dummies(df_label,
columns = ['City'])
Print('\n One-Hot Encoded Data (City)')
Print(df_encoded)
```

Subject : _____ Date : 13 | 9 | 25

Feature Scaling : It is the process of transforming numerical features so they share a similar range or distribution.

Min-Max scaling (Normalization) : $x' = (x - Min) / (max - min)$, maps to $[0, 1]$.

Spilliting data into two parts

i) One part will use to train model

ii) Another part for testing

↓ ↓

import pandas as pd

from sklearn.Preprocessing import

Standard Scaler, MinMax Scaler

from sklearn.model_selection import

train_test_split

HEART'S

Subject : Date :

Standard Scaler, $z = \frac{x - \mu}{\sigma}$

x = Actual value, μ = mean

σ = Standard deviation of a column.

MinMax Scaled = $\frac{x - x_{min}}{x_{max} - x_{min}}$

x = Actual value, x_{max} = Max value of that column, x_{min} = Min value of that column.

train_test_split

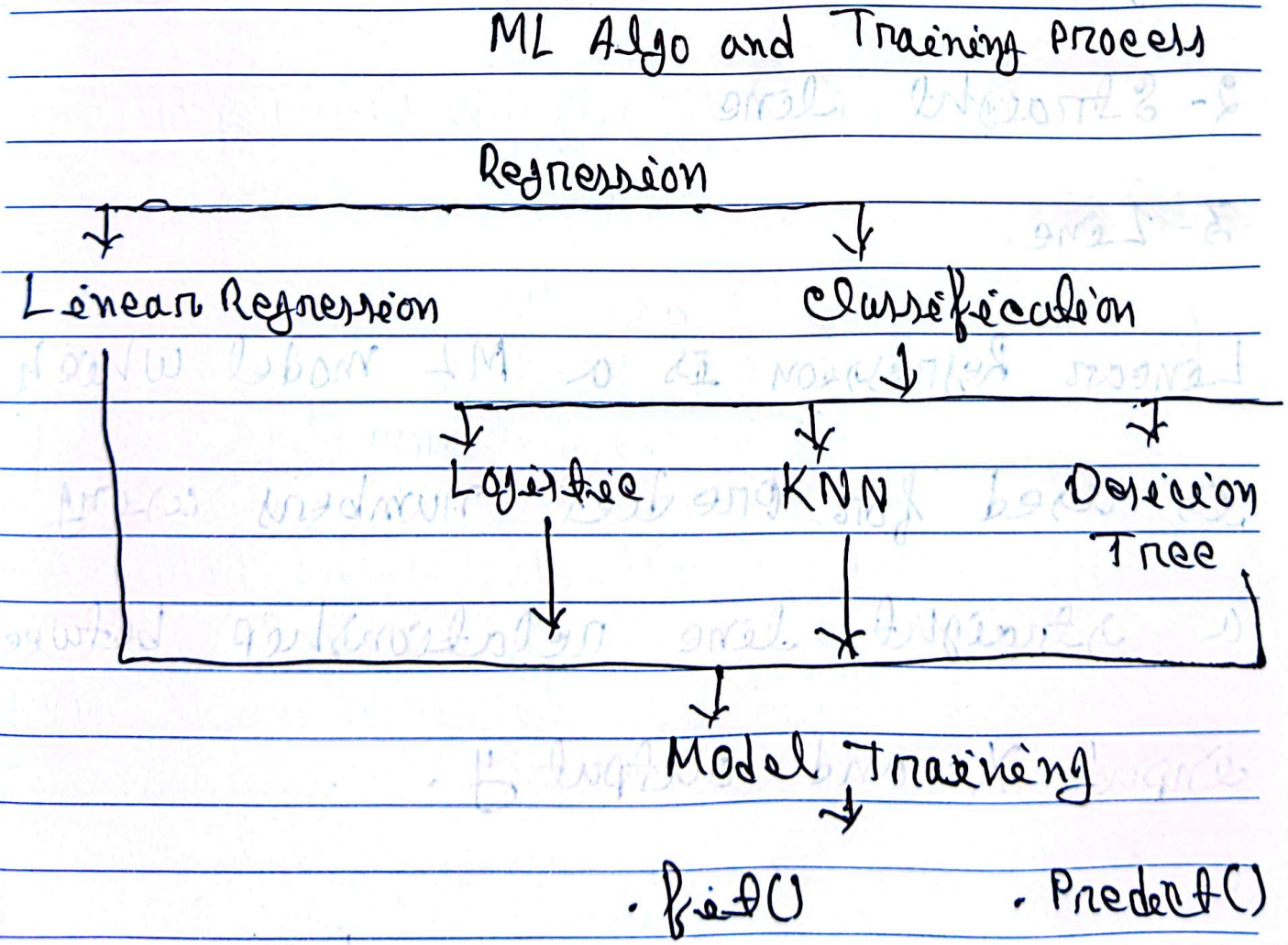
Subject : _____ Date : 14 | 9 | 2025

Phase 3: Supervised ML

Supervised means we will train model with output.

i) Predict numbers ii) Categories Predict

Collect label data → Train Model → Make Predictions → Evaluate performance → Refine Model.



Subject: Date: / /

We use regression when we have to Predict numbers. Ex: Marks, Price

We use classification when we have to predict a category. Ex: Mail spam.

Linear Regression:-

1- finds a pattern

2- Straight line

3- Line.

Linear Regression is a ML model which is used for predict numbers using a straight line relationship between input X and output y.

Subject : Date : 15 9 25

Classification is about predicting a discrete label (category), not a continuous number like Linear Regression.

Ex: Pass vs Fail, Spam vs not spam.

Logistic Regression is a ML model used for predicting a binary (0,1 / Yes, No) output.

It is a classification model unlike linear regression.

```
from sklearn.linear_model import LogisticRegression
X=[[1],[2],[3][4],[5]] #hours studied
y=[0,0,1,1,1] #0 is fail, 1 is pass
model=LogisticRegression()
model.fit(X,y)
hours=float(input("Enter Hours:"))
```

HEART'S

Subject : Date : 16 9

result = model.Predict([[Thours]])[0]

if result == 1:

Print("You are likely to pass")

else:

Print("You may fail")

KNN - K nearest neighbours

KNN is a simple, instance based supervised ML algorithm used for classification.

i) Chose K: Select the number of neighbors (K) ex: K = 3 (use odd number)

ii) Calculate distance: For a given query point, calculate the distance to all points in the training

Subject : _____ Date : 16 | 9 | _____

iii) Find Neighbors: Identify the K points closest to the query point.

iv) Vote: The query point is assigned to the class most common among its K nearest neighbors.

Taking decision based on nearest data point.

Suppose, in your friend group, 7 friends likes Football. So, there is a possibility that 8th friend can stand liking football.

KNN is not suitable for big dataset.

HEART

Subject: Fruit Example

Date: --/--/----

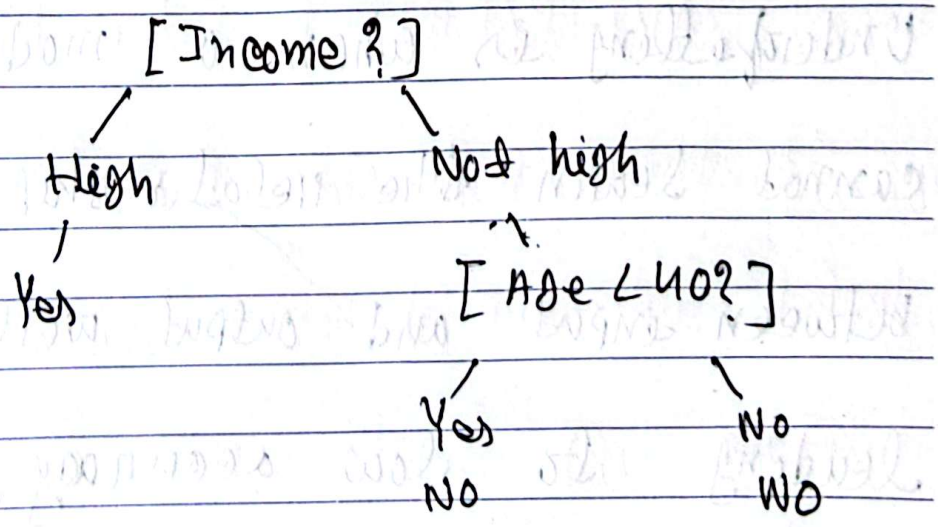
Weight(gm)	Size(cm)	Fruit
180	7	Apple
200	7.5	11
250	8	11
300	8.5	orange
320	9	11
360	9.5	11

```
from sklearn.neighbors import KNN
KNeighborsClassifier
X = [ [ ] # some input ]
y = [ #output based on input ]
model = KNeighborsClassifier(n_neighbors = 3)
model.fit(# X, y)
# Take value of X from user
prediction = model.Predict([[X]])[0]
# Print answer
```

Subject : _____ Date : __ __ ____

Decision Tree

A decision tree is a supervised ML algorithm used for classification and regression tasks. It works ~~like~~ by splitting data into branches based on feature values, creating a tree like model of decisions.



age	Income	buys a sports car
25	High	Yes
40	low	No
30	High	Yes
50	Medium	NO
35	Low	NO

HEART

Subject : _____ Date : __ __

Overfitting :- Overfitting is when a ML model learns the training data too closely and loses the ability to generalize to new data. (Gives poor performance on unseen examples)

Underfitting is when a model cannot learn the relationship between input and output well enough, leading to low accuracy.

Perfect fit : Perfect and accurate for model.

Subject : Date : 17 9 2025

Phase 4: Model Evaluation and METRICS

SKlearn Metrics: Master the use of 'sklearn.metrics' for comprehensive model evaluation.

Regression Metrics: Learn to evaluate regression models MAE, MSE and RMSE

Confusion Metrix:

Classification Metrics:-

Accuracy:- (%) of total right prediction

Acc = currect_pred / total_pred x 100

Precision: It measures how many of the items predicted as positives are actually positive. Precision = TP / (TP+FP)

TP = True positives, FP = False positives.

HEART

Subject: _____ Date: __ __ __

Ex: If a spam folder detects 10 emails as spams and 8 of them are actual spam, the precision is 8/10 = 0.8 (80%)

Recall: Recall measures how many of the actual positive items were correctly identified by the model.

Recall = TP / (TP/FN) FN = False Negative

Ex: If there are 12 spam emails in total, and the filter correctly detects 8 of them, recall is 8/12 = 0.67 (67%)

F1 score: It is a metric used in ML to evaluate the performance of a classification model by

Subject : _____ Date : __ / __ / ____

Combining both Precision and Recall into a single number.

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Range: 0 - 1. It is useful when we need to balance False Positives and False negatives and when dataset is imbalanced.

Subject : _____ Date : __/__/____

Confusion Matrix: A confusion matrix is a table used in ML to evaluate the performance of a classification algorithm. It shows how many predictions were correct and incorrect, and breaks them down by each class.

Actual / Predicted	Spam	Not spam
Spam	3	1
Not spam	2	4

MAE, MSE, RMSE

Student	Real Mark	Model Mark	Mes Pd
A	90	85	5
B	60	70	10
C	80	70	10
D	100	95	5

Subject : Date :

i) MAE (Mean Absolute Absolute Error) :

The average of the absolute differences between actual and predicted values.

$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$ y_i is actual value, $\hat{y}_1$ is the predicted value.

1) Take the mistake difference

2) Remove the minus sign

3) add, 4) divide

ii) MSE (Mean Squared Error) :- The average of the squared differences between actual and predicted values.

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

Useful when large error are particularly undesireable.

HEART'S

Subject : _____ Date : 7 - 7 - 2

1) Square mistakes

2) Add

3) Divide by N

RMSE (Root Mean Squared Error) : Just the square root of MSE

Useful : Penalizes large errors more than MAE. These 3 are most important.

Subject : _____ Date : -- - --

3 Step Formula:-

1- X and y Numeric ?

Scatter Plot Relationships

2- Is one column a category

Bar Plot / Count Plot

3- Want to see a distribution ?

Histogram / KDE Plot / Box Plot

Problem statement: Final Exam Score