IDS

Points	A1(2,10) cluster01	A4(5,8) cluster02	A7(1,2) cluster03	Belong to
A1 (2,10)	0	5	9	C1
A2 (2,5)	5	6	4	C3
A3 (8,4)	12	7	9	C2
A4 (5,8)	5	0	10	C2
A5 (7,5)	10	5	9	C2
A6 (6,4)	10	5	7	C2
A7 (1,2)	9	10	0	C3
A8 (4,9)	3	2	10	C2

new cluster 01 = (2,10)

new cluster 02 = $\frac{(8+5+7+6+4)}{5}, \frac{(4+8+5+4+9)}{5} = (6,6)$

new cluster 03 = $\frac{(2+1)}{2}, \frac{(5+2)}{2} = \left(\frac{3}{2}, \frac{7}{2}\right) = (1.5, 3.5)$

2nd :

Points	A1(2,10) cluster01	A4(6,6) cluster02	A7(1.5,3.5) cluster03	Belong to
A1 (2,10)	0	8	7	C1
A2 (2,5)	5	5	2	C3
A3 (8,4)	12	4	7	C2
A4 (5,8)	5	3	8	C2
A5 (7,5)	10	2	7	C2
A6 (6,4)	10	2	5	C2
A7 (1,2)	9	9	2	C3
A8 (4,9)	3	5	8	C1

new cluster1 = $\frac{(2+4)}{2}, \frac{(10+9)}{2}, = (3, 9.5)$

new cluster 2 = $\frac{(8+5+7+6)}{4}, \frac{(4+8+5+4)}{4} = (6.5, 5.25)$

new cluster 3 = $\frac{(2+1)}{2}, \frac{(5+2)}{2} = (1.5, 3.5)$

3rd :

Points	(3,9.5) cluster 01	(6.5,5.25) cluster 02	(1.5,3.5) cluster 03	Belongto.
A1 (2,10)	1.5	9.25	7	C1
A2 (2,5)	5.5	4.75	2	C3
A3 (8,4)	10.5	2.75	7	C2
A4 (5,8)	3.5	4.25	8	C1
A5 (7,5)	8.5	0.75	7	C2
A6 (6,4)	8.5	1.75	5	C2
A7 (1,2)	9.5	8.75	2	C3
A8 (4,9)	1.5	6.25	8	C1

CLUSTERING

individual	variable 1	variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

cluster	variable 1	variable 2
K1	1.0	1.0
K2	5.0	7.0

- distances from cluster 1 (1.0, 1.0) — cluster 2 (5.0, 7.0)

$(1.0, 1.0) = \sqrt{(1.0-1.0)^2 + (1.0-1.0)^2} = 0$ — $= 7.21$

$(1.5, 2.0) = \sqrt{(1.5-1.0)^2 + (2.0-1.0)^2} = 1.11$ — $= 6.10$

$(3.0, 4.0) = \sqrt{(3.0-1.0)^2 + (4.0-1.0)^2} = 3.60$ — $= 3.61$

$(5.0, 7.0) = \sqrt{(5.0-1.0)^2 + (7.0-1.0)^2} = 7.21$ — $= 0$

$(3.5, 5.0) = \sqrt{(3.5-1.0)^2 + (5.0-1.0)^2} = 4.71$ — $= 2.5$

$(4.5, 5.0) = \sqrt{(4.5-1.0)^2 + (5.0-1.0)^2} = 5.32$ — $= 2.06$

$(3.5, 4.5) = \sqrt{(3.5-1.0)^2 + (4.5-1.0)^2} \cdot 4.30$ — $= 2.92$

- new centroids

$$C1 = \frac{1}{3}(1.0 + 1.5 + 3.0), \frac{1}{3}(1.0 + 2.0 + 4.0) = (1.83, 2.33)$$

$$C2 = \frac{1}{4}(5.0 + 3.5 + 4.5 + 3.5), \frac{1}{4}(7.0 + 5.0 + 5.0 + 4.5) = (4.12, 5.38)$$

individual	centroid ~~variable~~ 1	centroid 2
1	0	7.21
2	1.11	6.10
3	3.60	3.61
4	7.21	0
5	4.71	2.5
6	5.32	2.06
7	4.30	2.92

individual	centroid 1	centroid 2	
1	1.57	~~0.47~~ 5.37	1
	0.47	4.27	2
	2.03	1.77	2
	5.64	1.84	2
	3.14	0.727	2
	3.77	0.53	2
	2.74	1.08	2

– ADVANTAG...

* at first calculate distances from the selected cluster points and classify them in clusters according to the minimum points.
* then find the new centroid and calculate distances again. do this until the clusters are classified.

ADVANTAGES

* easy to represent
* can work in multiple dimensions
* depends on initial value.
* scales to large datasets.

DISADVANTAGES

* time - consuming to find optimal number of clusters.
* feature must be numeric and normalized.

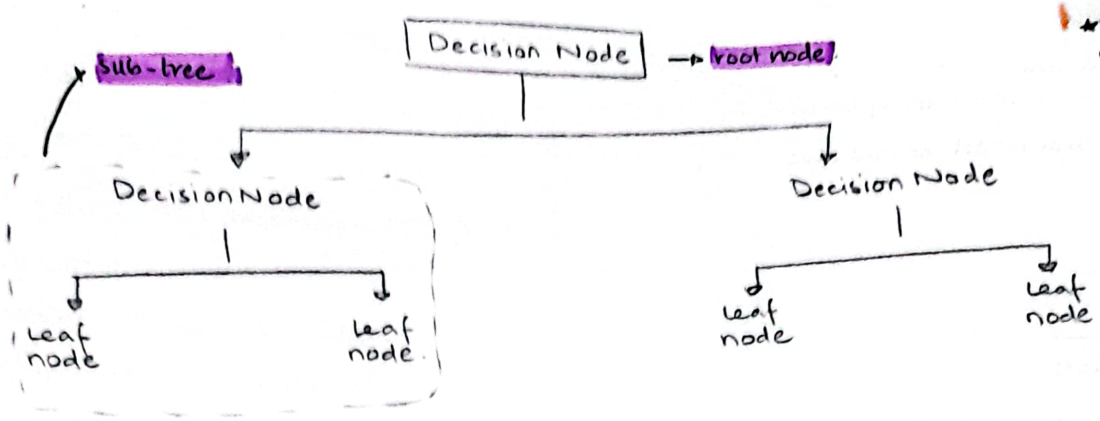
MACHINE LEARNING REGRESSION ANALYSIS

* prediction and forecasting.
* linear regression model

  $Y = b_0 + b_1 X + \varepsilon$ ∴ b_0 = yintercept b_1 = slop ε = error vaiable
* co-efficient of determination → R^2 .score() → r_sq = model.score(x,y)
* applications → economic growth, product price, housing sales, score predictions.

DECISION TREE ALGORITHM

* recursive partioning.



* internal node represents a feature/attribute.
* branch represents decision rule.
* leaf node represents outcome.

* attribute selection measures are Information gain, gain score, gini index

// split dataset into feature and target variables.
feature columns are independent
target columns are dependent.

// dividing dataset into training set and test set.
3 parameters are feature, target and test_set size.

// train decision tree   y_pred = clf.predict(x_test)

// predict response and accuracy (actual test set and predicted values)
metrics.accuracy_score(y_test, y_pred)

PROS

* easy to interpret and visualize.
* fewer data preprocessing.
* suitable for feature engineering.
* no assumptions.

CONS

* sensitive to noisy data.
* small variation result in different data tree.
* biased with imbalanced dataset.

MACHINE LEARNING PERFORMANCE METRICS

CONFUSION MATRIX

* correctness and accuracy.
* output can be of two or more type of classes
* ideal scenario is that model should give.
  0 FP and 0 FN

		Actual	
		Positive	Negative.
Predicted.	Positive	TP	FP
	Negative	FN	TN

ACCURACY

* $A = \frac{TP + TN}{TP+TN+FP+FN}$
* target variable classes are nearly balanced.

PRECISION

* $P = \frac{TP}{TP+FP}$

* ## RECALL/SENSITIVITY

* $R = \frac{TP}{TP+FN}$

* precision is about being precise.
* if focus is on minimizing FN, Recall should be as close to 100%.
* if focus is on minimizing FP, Precision should be as close to 100%.

F1 SCORE

* $\frac{2\,(\text{Precision})\,(\text{Recall})}{\text{Precision} + \text{Recall}}$

* if one number is really small between precision and recall, F1 score is more closer to smaller number.

SPECIFICITY

$= \frac{TN}{TN+FP}$

AREA UNDER ROC CURVE

True Positive Rate = Recall = $\frac{TP}{TP+FN}$

False Positive Rate = $\frac{FP}{FP+TN}$

* ranges from 0 to 1
  100% wrong → AUC = 0.0
  100% correct → AUC = 1.0

Mean Absolute Error → average of difference between original and predicted.
Mean Squared Error → average of square of difference between original & predicted.
Mean Absolute % Error → difference between actual & predicted / actual value, mean is taken.

...RATORY DATA ANALYSIS

- ...umerical methods and graphical tools.
- ...ploring data for pattern, trends
- ...maximize insight
- extract variables
- detect outliers and anomalies.
- classification
  - ↳ non-graphical : statistics
  - ↳ graphical : pictorial
  - ↳ univariate : one variable column.
  - ↳ multivariate : two or more variables.
- data types
  - ↳ categorical : nominal (no rank)
    ordinal (order)
  - ↳ numerical : discrete (counted)
    continuous (measured)

- high SD, scores are spread out
- low SD, scores near to mean.

COMPUTER VISION/IMAGE PROCESSING

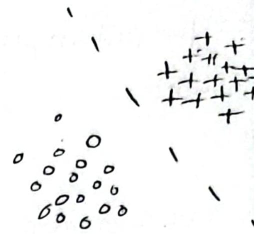
- image blurring: reduce noise or smooth out sharp edges
- applications
  - ↳ image classification
  - ↳ object detention
  - ↳ face recognition
  - ↳ face generation
  - ↳ image inpainting
  - ↳ text to image
  - ↳ optical character recognition

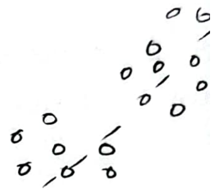
- cv2.imread
- edge detection : keeps important structural feature.
- sobel and prewitt : high frequency signals go through.
- canny edge : highest accuracy.

SUPERVISED LEARNING

- labelled data
  - ↳ classification : predict y labels (classes) for input x , discrete
    - spam detection, OCR, medical diagnosis , fraud detection.
    - image classification , customer retention
  - ↳ regression : predict continuous valued output.
    - predicting price , income, age , weather prediction
    - population growth, market forecasting , life expectancy

classificat





UNSUPERVISED LEARNING

- unlabelled data.
- finds hidden pattern
- only input is provided
  - ↳ clustering
    - targeted marketing , customer segmentation
    - detection anomalies.

TEXT PREPROCESSING IN NLP

- normalization

```
sentence = sentence.lower()
```

- tokenization

```
sentences = nltk.sent_tokenize(text)
words = nltk.word_tokenize(sentence)
```

- stemming
  - ↳ finds common base root words
  - ↳ chops start or end.
- lemmatization
  - ↳ existing word.
- unigrams

```
unigrams = nltk.ngrams(tokens, 1)
bigrams = nltk.ngrams (tokens, 2)
trigrams = nltk.ngrams (tokens, 3)
FD = nltk.FreqDist(text)
```

CHAPTER 6

-ADVANTAGES OF [illegible]

1	2	3
4	5	6

a = np.array([1,2,3,4])

b = np.array([(1,2,3),(4,5,6)], dtype=float)

np.zeros((3,4)) → 3 rows, 4 columns

np.ones((2,3,4)) → 2 arrays, 3 rows, 4 columns

5. np.random.random((2,2)) → 2x2 by array

6. a[0:2] → array([1,2])

7. b[0:2,1] → items at rows 0 and 1 in column 1

MATPLOTLIB

1. fig.savefig → save image
2. plt.legend → linestyle and color, matches with correct label
3. alpha → transparency level
4. clf → clear entire file
5. cla → clear entire axis

loc (by label)

- df.loc[0:1] → 0 and 1 are included
- df.iloc[0:1] → 1 not included

PANDAS

1. pd.read_csv
2. pd.read_excel.
3. df.head() → returns first five rows.
4. df.shape() → returns a tuple with number of rows & column.
5. df.describe() → mean, std
6. df.columns() → name of all columns in a list.
7. df.index → index of dataframe.
8. df.dtypes
9. df.isna().sum() → how many missing values in each column.
10. df['columnname'].replace('70-80', 0.7)
11. df['columnname'].isin(some_list) → returns true or false.
12. df.groupby('columnname')
13. df.iloc([0], [0]) → select value by row and column.
14. df.loc([0], ['Country']) → select by column labels.
15. df.drop('column name'), axis=1)
16. df.drop(number) → axis=0, drops row.
17. df.dropna() → rows with missing value is removed.
18. df.drop_duplicates → rows with duplicate value is removed.
19. df.drop_duplicates(['columnname']) → check for duplicates in given column.
20. df.fillna(value) → missing values are replaced.
21. df.info → number of columns, labels, datatypes, range index.
22. df.count → number of non null values.
23. pd.concat → adding rows from two dataframe.
24. df.merge(df2, how='left', on='columnname') → column taken as reference.

list = [ ]
tuples = ( )
dictionary = { }

25. get_dummies() → categorical variables into dummy/indicator