For all GATE Data Science & Artificial Intelligence preparing students, hope this PPT will be useful. Contents taken from various sources.

Presentation · October 2023

CITATIONS
0

READS
340

1 author:

Arun Anoop Mandankandy
Vel Tech - Technical University
53 PUBLICATIONS **72** CITATIONS

SEE PROFILE

GRADUATE APTITUDE TEST IN ENGINEERING 2024

अभियांत्रिकी स्नातक अभिक्षमता परीक्षा २०२४

ORGANISING INSTITUTE: INDIAN INSTITUTE OF SCIENCE, BENGALURU

GATE New Test Paper on (DA) Data Science and Artificial Intelligence

Syllabus

Dr Arun Anoop M

Associate Professor

Dept. of CSE

Veltech Technical University.

GATE 2024 IISc Bengaluru

GRADUATE APTITUDE TEST IN ENGINEERING 2024

अभियांत्रिकी स्नातक अभिक्षमता परीक्षा २०२४

ORGANISING INSTITUTE: INDIAN INSTITUTE OF SCIENCE, BENGALURU

GATE New Test Paper on (DA) Data Science and Artificial Intelligence

Syllabus

Machine Learning: (i) Supervised Learning: regression and classification problems, simple linear regression, multiple linear regression, ridge regression, logistic regression, k-nearest neighbour, naive Bayes classifier, linear discriminant analysis, support vector machine, decision trees, bias-variance trade-off, cross-validation methods such as leave-one-out (LOO) cross-validation, k-folds cross-validation, multi-layer perceptron, feed-forward neural network; (ii) Unsupervised Learning: clustering algorithms, k-means/k-medoid, hierarchical clustering, top-down, bottom-up: single-linkage, multiple-linkage, dimensionality reduction, principal component analysis.

Supervised Learning:

- Supervised learning is a machine learning approach where the model learns from labeled training data to make predictions or classify new, unseen data. It involves two types of problems:

1. Regression Problems: In regression, the goal is to predict a continuous target variable. Some commonly used regression algorithms include:

- Simple Linear Regression: It models the relationship between a single input feature and a continuous target variable using a linear equation.
- Multiple Linear Regression: It extends simple linear regression to multiple input features.
- Ridge Regression: It is a regularized version of linear regression that adds a penalty term to control the complexity of the model and reduce overfitting.
- Logistic Regression: It models the relationship between input features and the probability of belonging to a specific class. It is commonly used for binary classification problems.

Regression Problems

- **Regression** aims to estimate a continuous target variable based on input features.
- Common examples of regression problems include:

1. Predicting House Prices: Given features like the number of bedrooms, square footage, and location, the goal is to estimate the price of a house.
2. Stock Market Prediction: Using historical data, predict the future price or return of a stock.
3. Sales Forecasting: Based on factors like advertising expenditure, seasonality, and economic indicators, predict the sales volume of a product.
4. Temperature Prediction: Given historical weather data, predict the temperature for a future date.

	Regression	Classification
Description	A a regression model seeks to predict a continuous quantity.	A classification model seeks to predict some class label.
Type of algorithm	Supervised learning algorithm	Supervised learning algorithm
Type of response variable	Continuous	Categorial
How to assess model fit	Root mean squared error	Percentage of correct classifications

Evaluation of a regression model

- The evaluation of a regression model is typically done using metrics such as mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), or R-squared (coefficient of determination). These metrics quantify the performance of the model and help assess how well it can predict the target variable.

1. **Simple linear regression**

   Assume that there is only one independent variable x. If the relation between x and y is modeled by the relation

   $$y = a + bx$$

   then we have a simple linear regression.

2. **Multiple regression**

   Let there be more than one independent variable, say $x_1, x_2, \ldots, x_n$, and let the relation between y and the independent variables be modeled as

   $$y = \alpha_0 + \alpha_1 x_1 + \cdots + \alpha_n x_n$$

   then it is case of multiple linear regression or multiple regression.

3. **Polynomial regression**

   Let there be only one variable x and let the relation between x y be modeled as

   $$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

   for some positive integer $n > 1$, then we have a polynomial regression.

4. **Logistic regression**

   Logistic regression is used when the dependent variable is binary (0/1, True/False, Yes/No) in nature. Even though the output is a binary variable, what is being sought is a probability function which may take any value from 0 to 1.

enjoy algorithms

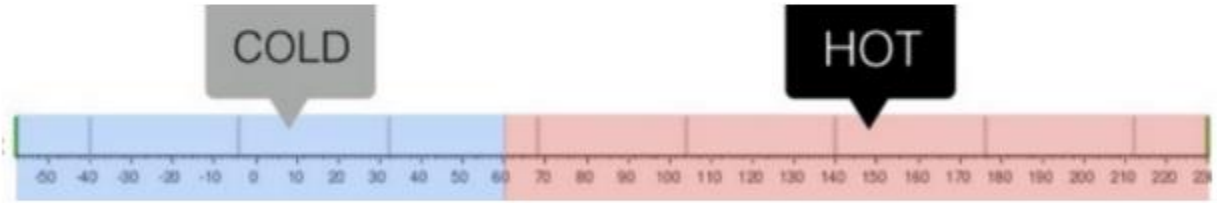
Regression

What will be the temperature tomorrow?

Fahrenheit

Classification

Will it be hot or cold tomorrow?

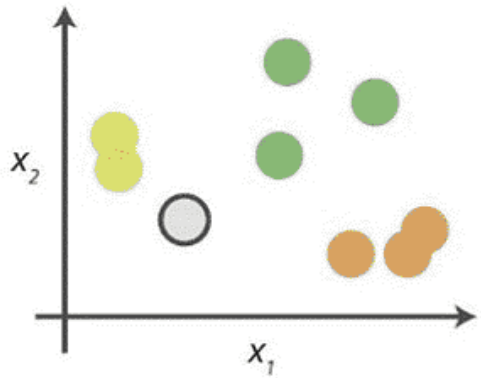


Fahrenheit

1. Classification Problems: In classification, the goal is to assign input data to predefined categories or classes. Some commonly used classification algorithms include:

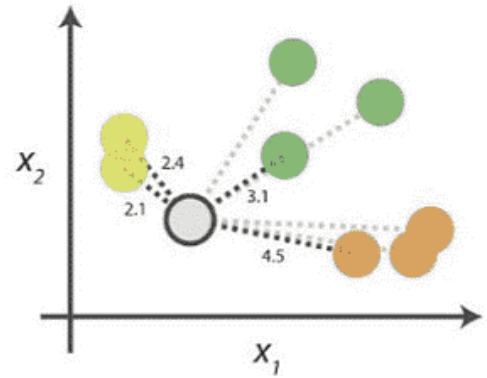
- k-Nearest Neighbors (k-NN): It classifies new instances based on the majority vote of k nearest neighbors in the training data.
- Naive Bayes Classifier: It applies Bayes' theorem with the assumption of independence between features to predict the class probabilities.

0. Look at the data



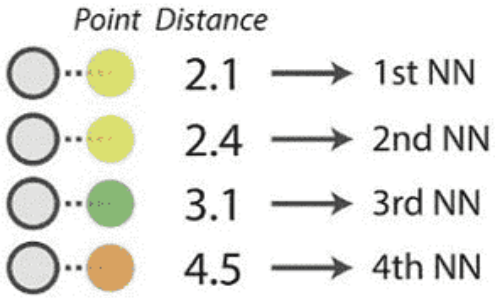
Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances



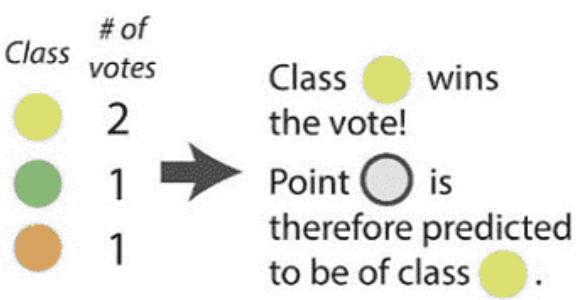
Start by calculating the distances between the grey point and all other points.

2. Find neighbours



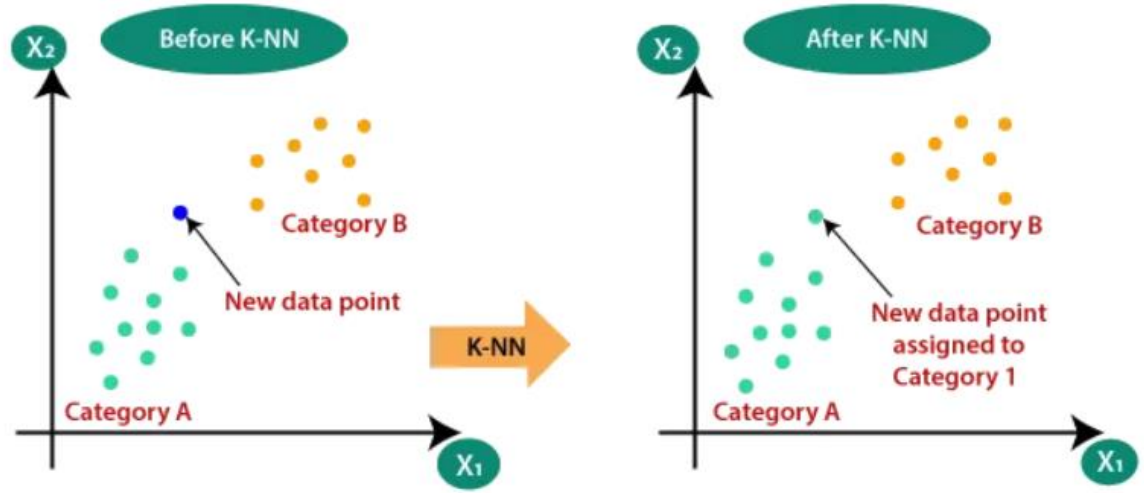
Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels



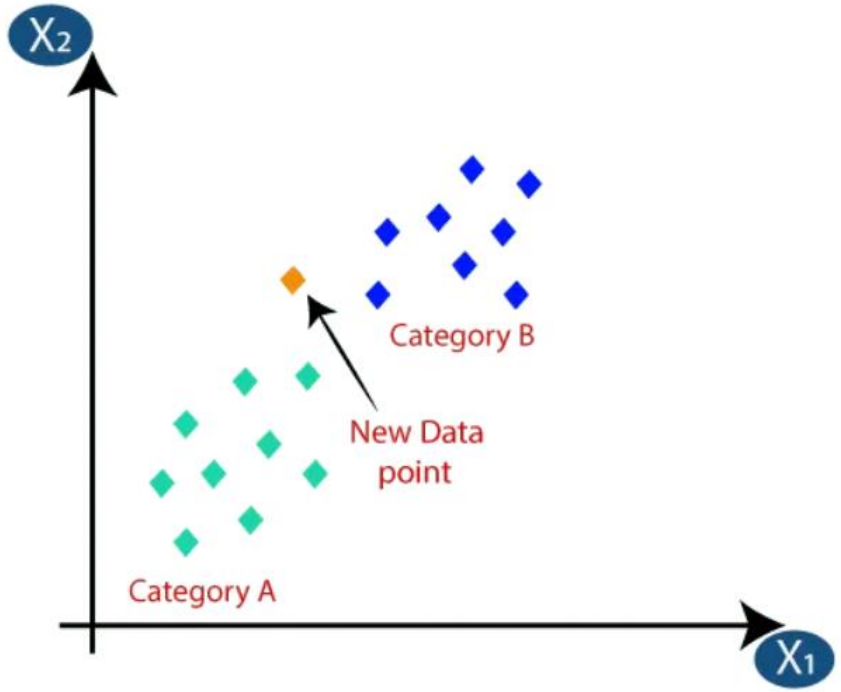
Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

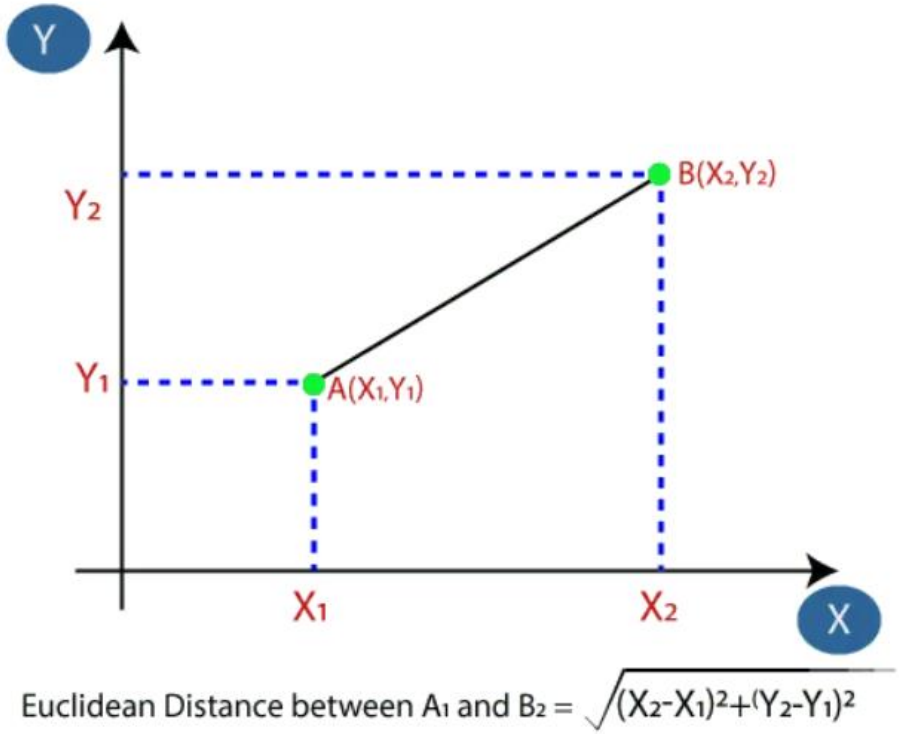
KNN Classifier



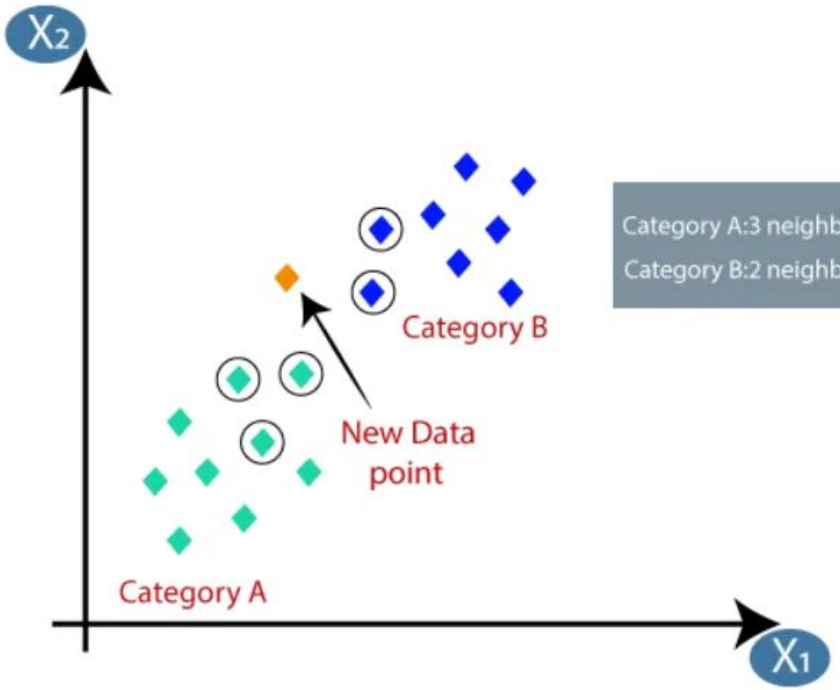
The K-NN working can be explained on the basis of the below algorithm:

- Step-1: Select the number K of the neighbors
- Step-2: Calculate the Euclidean distance of K number of neighbors
- Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
- Step-4: Among these k neighbors, count the number of the data points in each category.
- Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
- Step-6: Our model is ready.





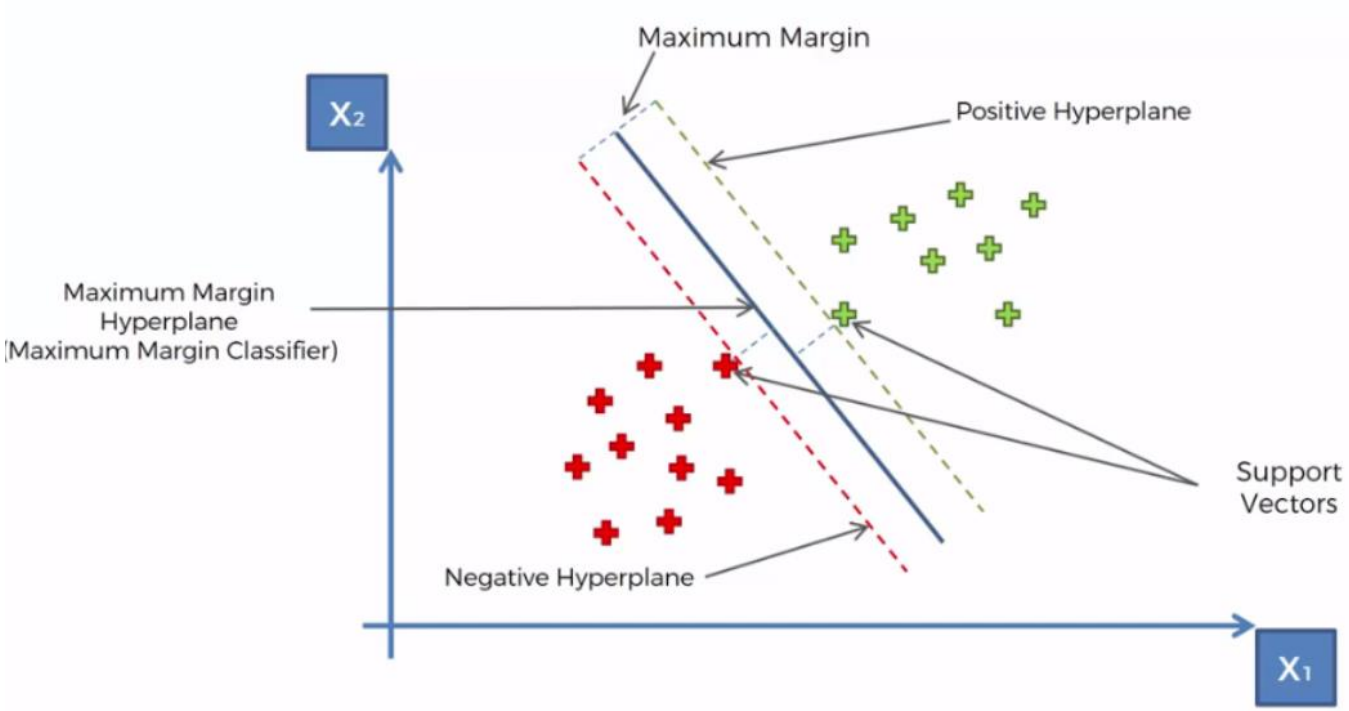
Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$



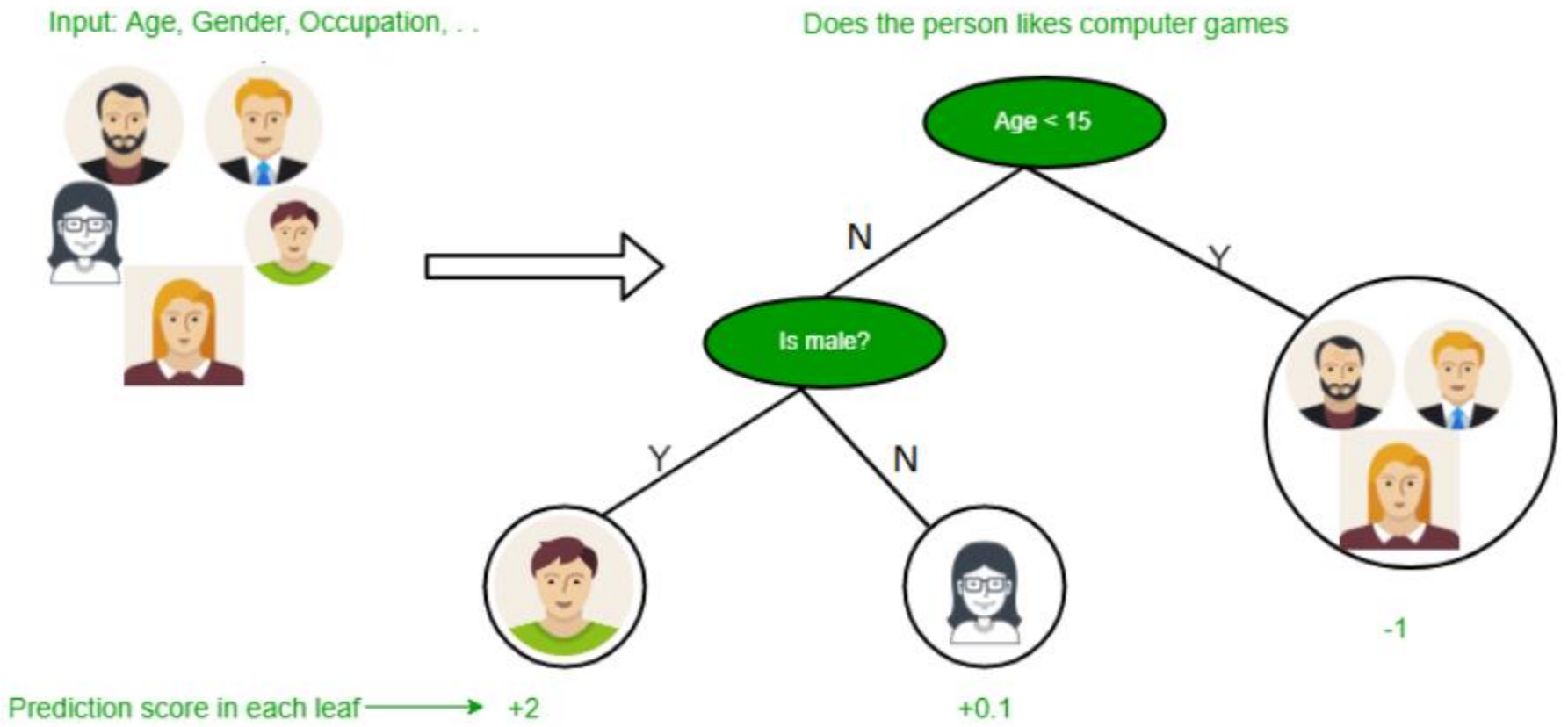
Category A:3 neighbors
Category B:2 neighbors

- Linear Discriminant Analysis (LDA): Linear discriminant analysis (LDA) is used to reduce the number of features.
- Support Vector Machine (SVM): SVM is used to **classify data by finding the optimal decision boundary that maximally separates** different classes.

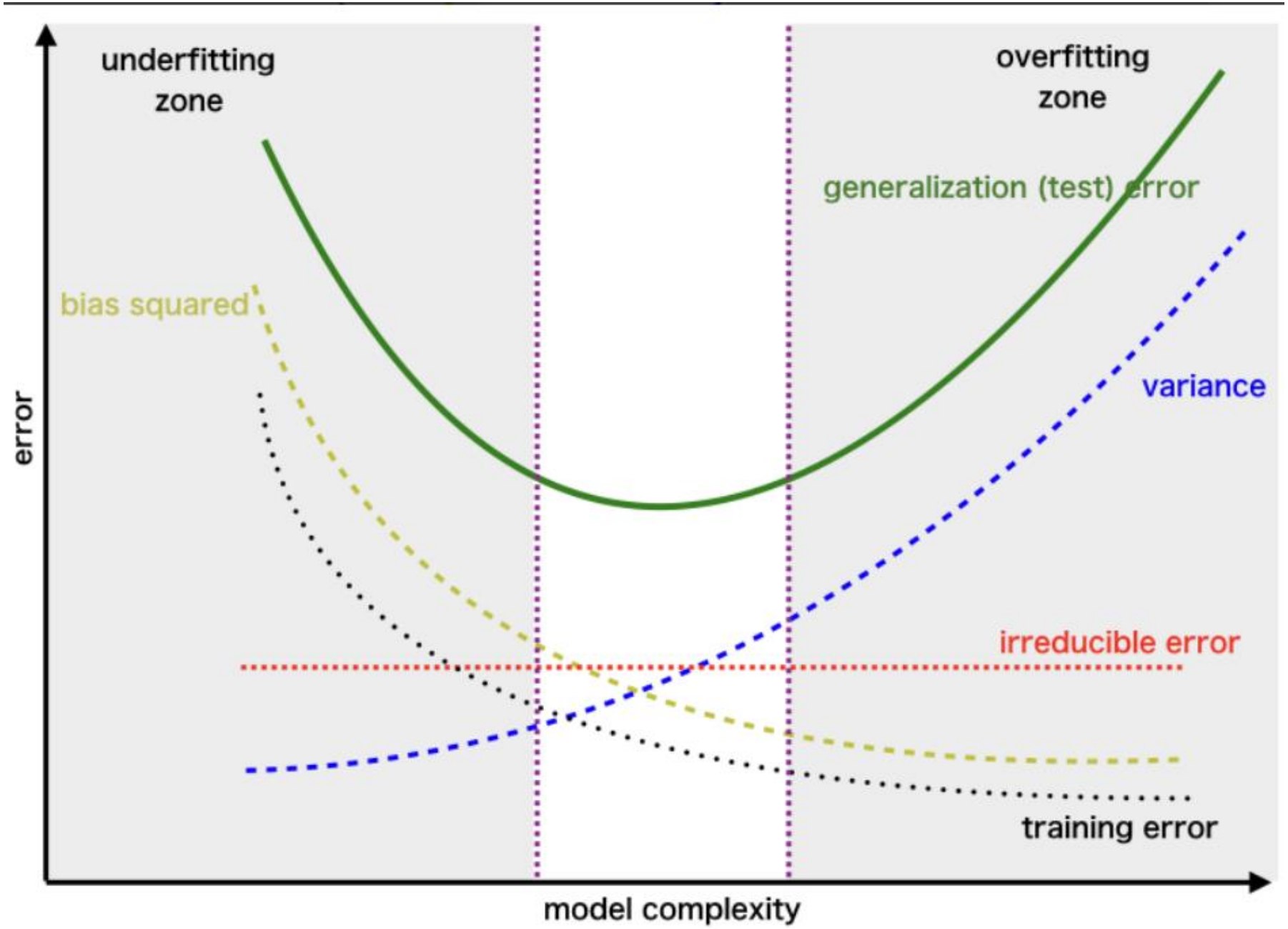
LDA
Supervised Method Requires labeled data Generates features up to # of labels-1 Appropriate for classification Generate axis based on classification performance



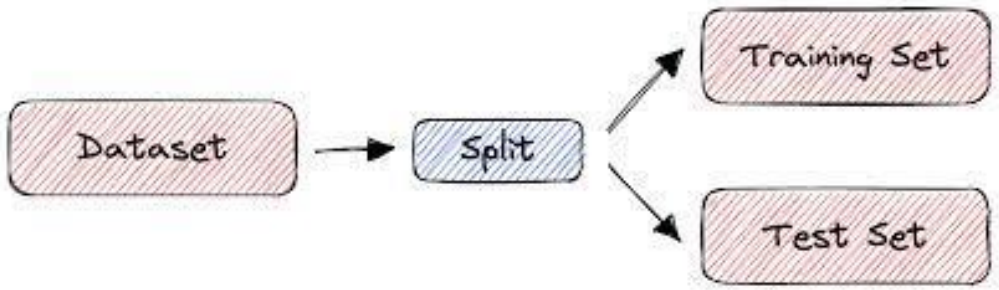
•Decision Trees: They recursively split the input space based on different features to create a tree-like model for classification.



- **Bias-Variance Trade-off**: In machine learning, the bias-variance trade-off refers to the relationship between the bias and variance of a model.
- Bias measures the model's ability to approximate the true underlying relationship between features and target variables, while variance measures the model's sensitivity to fluctuations in the training data.
- A model with high bias underfits the data by oversimplifying the relationships, while a model with high variance overfits the data by capturing noise and inconsistencies.
- The goal is to strike a balance between bias and variance to achieve optimal model performance.


underfitting
zone
overfitting
zone
generalization (test) error
bias squared
variance
irreducible error
training error
error
model complexity

- **Cross-validation**: Cross-validation is a technique used to assess the performance and generalization ability of a machine learning model. It involves splitting the available dataset into training and validation subsets.
- The model is trained on the training subset and then evaluated on the validation subset.
- Cross-validation helps estimate how well the model will perform on unseen data and helps in tuning hyperparameters and model selection.



1.Leave-One-Out Cross-Validation (LOOCV):

a) LOOCV is a specific cross-validation method where the model is trained on all but one data point and then tested on the left-out data point. This process is repeated for each data point in the dataset.

b) LOOCV provides an unbiased estimate of model performance but can be computationally expensive for large datasets.

2. K-Fold Cross-Validation:

a) K-fold cross-validation involves splitting the dataset into K equally sized folds. The model is trained on K-1 folds and evaluated on the remaining fold. This process is repeated K times, with each fold serving as the validation set once.

b) The results are averaged to obtain an overall performance estimate. K-fold cross-validation strikes a balance between computational efficiency and performance estimation accuracy.

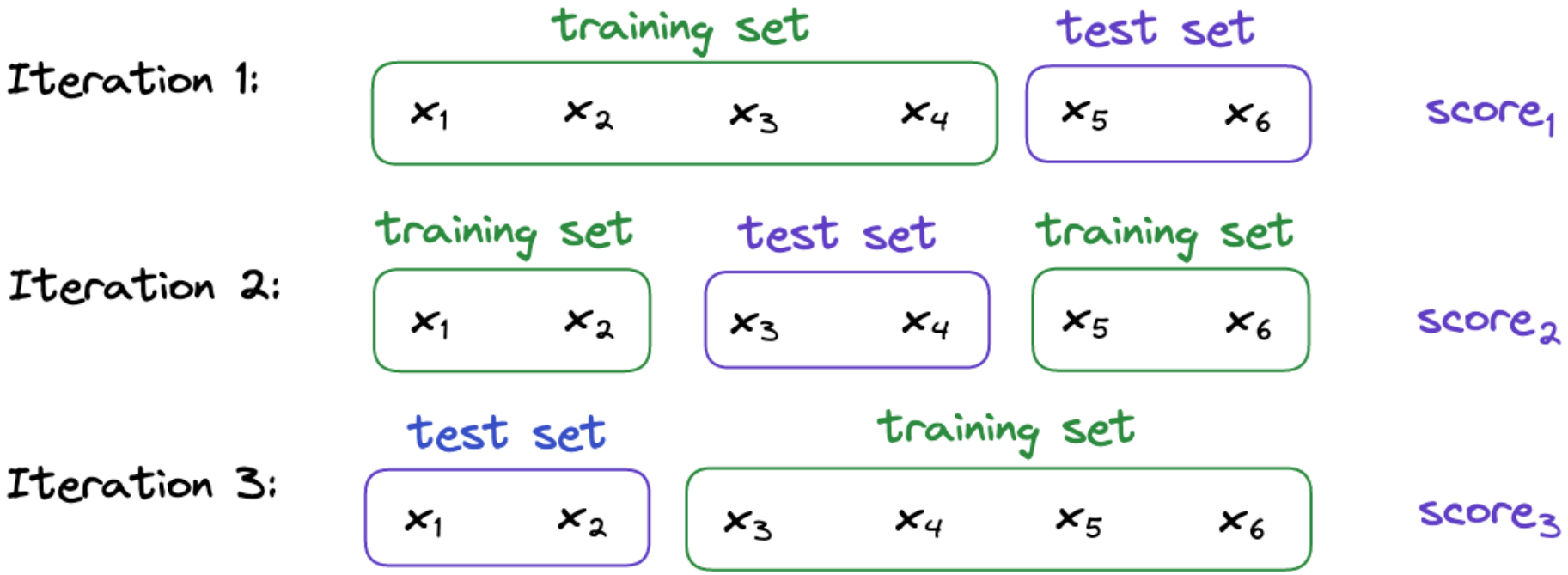
3. Jackknife CV (Jackknife Cross-Validation): Jackknife CV is similar to LOOCV but used for smaller datasets.

LOOCV

In the leave-one-out (LOO) cross-validation, we train our machine-learning model n times where n is to our dataset's size. **Each time, only one sample is used as a test set while the rest are used to train our model.**

The final performance estimate is the average of the six individual scores:

$$\text{overall score} = \frac{score_1 + score_2 + score_3 + score_4 + score_5 + score_6}{6}$$



K-Fold CV

In k-fold cross-validation, we first divide our dataset into k equally sized subsets. **Then, we repeat the train-test method k times such that each time one of the k subsets is used as a test set and the rest k-1 subsets are used together as a training set.** Finally, we compute the estimate of the model's performance estimate by averaging the scores over the k trials.

Then, we train and evaluate our machine-learning model 3 times. Each time, two subsets form the training set, while the remaining one acts as the test set. In our example:

$$\text{overall score} = \frac{score_1 + score_2 + score_3}{3}$$

- **Multi-Layer Perceptron (MLP) and Feed-Forward Neural Networks**: MLP is a type of feed-forward neural network, which is a popular architecture for supervised learning tasks. It consists of multiple layers of interconnected nodes (neurons).
- Each node applies a non-linear activation function to a weighted sum of its inputs. The layers include an input layer, one or more hidden layers, and an output layer.
- MLPs are trained using backpropagation, adjusting the weights to minimize the error between predicted and actual outputs.

Unsupervised Learning:

- Clustering Algorithms: Clustering algorithms are used in unsupervised learning to group similar instances together based on their characteristics. Some common clustering algorithms include:
- K-means/K-medoid:

✓K-means assigns instances to one of K clusters based on minimizing the within-cluster sum of squared distances.

✓K-medoid is a variant that uses representative medoids instead of means. Both algorithms require the number of clusters (K) to be specified in advance.

- Hierarchical Clustering: Hierarchical clustering builds a tree-like structure (dendrogram) to represent the relationships between instances. It can be performed in two ways:
  - Top-Down (Divisive): It starts with all instances in a single cluster and recursively splits them into smaller clusters until each instance forms its own cluster.
  - Bottom-Up (Agglomerative): It starts with each instance in its own cluster and recursively merges the most similar clusters until a single cluster is formed.

- Single-Linkage and Complete-Linkage: These are distance-based agglomerative hierarchical clustering methods. Single-linkage considers the minimum distance between instances in different clusters, while complete-linkage considers the maximum distance.
- Dimensionality Reduction: Principal Component Analysis (PCA): PCA is a popular dimensionality reduction technique used to reduce the number of input features while preserving most of the variability in the data.

- Dimensionality reduction is a technique used to reduce the number of input features in a dataset while retaining the most relevant information. It is commonly employed in machine learning to mitigate the risk of overfitting.
- One widely used dimensionality reduction technique is Principal Component Analysis (PCA). PCA transforms the original features into a new set of orthogonal variables called principal components.

Entropy of the set S

$$H(S) = -\sum_{c \in C} p_c \log_2 p_c$$

Probability vector $p = [p_1, p_2, \ldots, p_C]$ is the class distribution of the set S

Entropy is a measure of randomness/uncertainty of a set

Assume our data is a set S of examples with C many classes

p_c is the probability that a random element of S belongs to class c

Let's assume each element of S has a set of features

Information Gain (IG) on knowing the value of some feature 'F'

$$IG(S, F) = H(S) - \sum_{f \in F} \frac{|S_f|}{|S|} H(S_f)$$

S_f denotes the subset of elements of S for which feature F has value f

$IG(S, F)$ = entropy of S minus the weighted sum of entropy of its children

$IG(S, F)$: Increase in our certainty about S once we know the value of F

$IG(S, F)$ denotes the no. of bits saved while encoding S once we know the value of the feature F

Computing Information Gain

day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

wind	play
weak	no
strong	no
weak	yes
weak	yes
weak	yes
strong	no
strong	yes
weak	no
weak	yes
weak	yes
strong	yes
strong	yes
weak	yes
strong	no

- Root node: $S = [9+, 5-]$ (all training data: 9 play, 5 no-play)
- Entropy: $H(S) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.94$
- $S_{weak} = [6+, 2-] \Longrightarrow H(S_{weak}) = 0.811$
- $S_{strong} = [3+, 3-] \Longrightarrow H(S_{strong}) = 1$

Information Gain (IG) on knowing the value of some feature 'F'

$$IG(S, F) = H(S) - \sum_{f \in F} \frac{|S_f|}{|S|} H(S_f)$$

$$\begin{aligned} IG(S, \text{wind}) &= H(S) - \frac{|S_{weak}|}{|S|} H(S_{weak}) - \frac{|S_{strong}|}{|S|} H(S_{strong}) \\ &= 0.94 - 8/14 * 0.811 - 6/14 * 1 \\ &= 0.048 \end{aligned}$$

Name	Gives birth	Aquatic animal	Aerial animal	Has legs	Class label
human	yes	no	no	yes	mammal
bat	yes	no	yes	yes	bird
cat	yes	no	no	yes	mammal
shark	yes	yes	no	no	fish

Three class labels appear in this segment, namely, "bird", "fish" and "mammal".

Number of examples with class label "bird"	1
Number of examples with class label "fish"	1
Number of examples with class label "mammal"	2
Total number of examples	4

Therefore we have

$$\begin{aligned}
\text{Entropy}\,(S) &= \sum_{\text{for all classes "xxx"}} -p_{\text{xxx}} \log_2(p_{\text{xxx}}) \\
&= -p_{\text{bird}} \log_2(p_{\text{bird}}) - p_{\text{fish}} \log_2(p_{\text{fish}}) \\
&\quad - p_{\text{mammal}} \log_2(p_{\text{mammal}}) \\
&= -(1/4)\log_2(1/4) - (1/4)\log_2(1/4) - (2/4)\log_2(2/4) \\
&= -(1/4) \times (-2) - (1/4) \times (-2) - (2/4) \times (-1) \\
&= 1.5
\end{aligned}$$

Bayes, Thomas (1763) An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London,* **53:370-418**

Bayes Formula

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

$$P(B_k|A) = \frac{P(A|B_k)P(B_k)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)}$$

- At a certain university, 4% of men are over 6 feet tall and 1% of women are over 6 feet tall. The total student population is divided in the ratio 3:2 in favour of women. If a student is selected at random from among all those over six feet tall, what is the probability that the student is a woman?

- Find P(F|T)?
- P(F|T)=P(T|F)P(F)/P(T/F)P(F)+P(T/M)P(M)

- The proportions of bike owner at the petrol pump in the city using regular, extra unleaded and premium petrol are 40%, 35%, and 25% respectively. The respective proportions of filling their tanks are 30%, 50%, and 60%. If a randomly chosen motorist filled his/her tanks, what is the probability that he/she used regular petrol?

- Find P(R|F)?

- $$P(R|F)=P(F|R)P(R)/P(F|R)P(R)+P(F|E)P(E)+P(F|P)P(P)$$

If a classification system has been trained to distinguish between cats, dogs and rabbits, a confusion matrix will summarize the results of testing the algorithm for further inspection. Assuming a sample of 27 animals - 8 cats, 6 dogs, and 13 rabbits, the resulting confusion matrix could look like the table below: This confusion matrix shows that, for example, of the 8 actual cats, the system predicted that three were dogs, and of the six dogs, it predicted that one was a rabbit and two were cats.

Two-class datasets	Actual condition is true	Actual condition is false
Predicted condition is true	TP	FP
Predicted condition is false	FN	FN

Multiclass datasets

	Actual "cat"	Actual "dog"	Actual "rabbit"
Predicted "cat"	5	2	0
Predicted "dog"	3	3	2
Predicted " rabbit"	0	1	11

Problem 1

Suppose a computer program for recognizing dogs in photographs identifies eight dogs in a picture containing 12 dogs and some cats. Of the eight dogs identified, five actually are dogs while the rest are cats. Compute the precision and recall of the computer program.

Solution

We have:

$$\begin{aligned} \text{TP} &= 5 \\ \text{FP} &= 3 \quad \text{8-5=3} \\ \text{FN} &= 7 \quad \text{12-5=7} \end{aligned}$$

The *precision* P is

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{5}{5+3} = \frac{5}{8}$$

The *recall* R is

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{5}{5+7} = \frac{5}{12}$$

Problem 3

Assume the following: A database contains 80 records on a particular topic of which 55 are relevant to a certain investigation. A search was conducted on that topic and 50 records were retrieved. Of the 50 records retrieved, 40 were relevant. Construct the confusion matrix for the search and calculate the precision and recall scores for the search.

Solution

Each record may be assigned a class label "relevant" or "not relevant". All the 80 records were tested for relevance. The test classified 50 records as "relevant". But only 40 of them were actually relevant. Hence we have the following confusion matrix for the search:

	Actual "relevant"	Actual "not relevant"
Predicted "relevant"	40	10
Predicted "not relevant"	15	25

Table 5.2: Example for confusion matrix

$$\begin{aligned} \text{TP} &= 40 \\ \text{FP} &= 10 \\ \text{FN} &= 15 \end{aligned}$$

The *precision* P is

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{40}{40 + 10} = \frac{4}{5}$$

The *recall* R is

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{40}{40 + 15} = \frac{40}{55}$$

Problem 1

Consider a set of patients coming for treatment in a certain clinic. Let A denote the event that a "Patient has liver disease" and B the event that a "Patient is an alcoholic." It is known from experience that 10% of the patients entering the clinic have liver disease and 5% of the patients are alcoholics. Also, among those patients diagnosed with liver disease, 7% are alcoholics. Given that a patient is alcoholic, what is the probability that he will have liver disease?

Solution

Using the notations of probability, we have

$$P(A) = 10\% = 0.10$$
$$P(B) = 5\% = 0.05$$
$$P(B|A) = 7\% = 0.07$$
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$
$$= \frac{0.07 \times 0.10}{0.05}$$
$$= 0.14$$

Problem 2

Three factories A, B, C of an electric bulb manufacturing company produce respectively 35%. 35% and 30% of the total output. Approximately 1.5%, 1% and 2% of the bulbs produced by these factories are known to be defective. If a randomly selected bulb manufactured by the company was found to be defective, what is the probability that the bulb was manufactures in factory A?

Solution

Let A, B, C denote the events that a randomly selected bulb was manufactured in factory A, B, C respectively. Let D denote the event that a bulb is defective. We have the following data:

$$P(A) = 0.35, \quad P(B) = 0.35, \quad P(C) = 0.30$$
$$P(D|A) = 0.015, \quad P(D|B) = 0.010, \quad P(D|C) = 0.020$$

We are required to find $P(A|D)$. By the generalisation of the Bayes' theorem we have:

$$P(A|D) = \frac{P(D|A)P(A)}{P(D|A)P(A) + P(D|B)P(B) + P(D|C)P(C)}$$
$$= \frac{0.015 \times 0.35}{0.015 \times 0.35 + 0.010 \times 0.35 + 0.020 \times 0.30}$$
$$= 0.356.$$

Data

Nam	Features				Class label
	gives birth	aquatic animal	aerial animal	has legs	
human	yes	no	no	yes	mammal
python	no	no	no	no	reptile
salmon	no	yes	no	no	fish
frog	no	semi	no	yes	amphibian
bat	yes	no	yes	yes	bird
pigeon	no	no	yes	yes	bird
cat	yes	no	no	yes	mammal
shark	yes	yes	no	no	fish
turtle	no	semi	no	yes	amphibian
salamander	no	semi	no	yes	amphibian

Some well-known decision tree algorithms

ID3 (Iterative Dichotomiser 3) developed by Ross Quinlan

C4.5 developed by Ross Quinlan

C5.0 developed by Ross Quinlan

CART (Classification And Regression Trees)

1R (One Rule) developed by Robert Holte in 1993.

RIPPER (Repeated Incremental Pruning to Produce Error Reduction) Introduced in 1995 by William W. Cohen.

The Gini split index of a data set is another feature selection measure in the construction of classification trees. This measure is used in the CART algorithm.

Let S be the data in Table 8.1. There are four class labels "amphi", "bird", "fish", "mammal" and "reptile". The numbers of examples having these class labels are as follows:

Number of examples with class label "amphi"	= 3
Number of examples with class label "bird"	= 2
Number of examples with class label "fish"	= 2
Number of examples with class label "mammal"	= 2
Number of examples with class label "reptile"	= 1
Total number of examples	= 10

The Gini index of S is given by

$$\text{Gini}(S) = 1 - \sum_{i=1}^{r} p_i^2.$$

$$\begin{aligned}\text{Gini}(S) &= 1 - \sum_{i=1}^{r} p_i^2 \\ &= 1 - (3/10)^2 - (2/10)^2 - (2/10)^2 - (2/10)^2 - (1/10)^2 \\ &= 0.78\end{aligned}$$

The *gain ratio* is a third feature selection measure in the construction of classification trees.

Let S be a set of examples, A a feature having c different values and let the set of values of A be denoted by Values(A). We defined the information gain of A relative to S, denoted by Gain(S, A), by

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values(A)}} \frac{|S_v|}{|S|} \times \text{Entropy}(S_v).$$

We now define the*split information* of A relative to S, dented by SplitInformation(S, A), by

$$\text{SplitInformation}(S, A) = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

where $S_1, \ldots S_c$ are the c subsets of examples resulting from partitioning S into the c values of the attribute A. The *gain ratio* of A relative to S, denoted by GainRatio(S, A), by

$$\text{GainRatio}(S, A) = \frac{\text{Gain(S,A)}}{\text{SplitInformation}(S, A)}.$$

Consider the data S given in Table 8.1. Let A denote the attribute "gives birth".We have have already seen that

$$|S| = 10$$
$$\text{Entropy}\ (S) = 2.2464$$
$$\text{Gain}(S, A) = 0.5709$$

Now we have

$$\begin{aligned}\text{SplitInformation}(S, A) &= -\frac{|S_{\text{yes}}|}{|S|}\log_2\frac{|S_{\text{yes}}|}{|S|} - \frac{|S_{\text{no}}|}{|S|}\log_2\frac{|S_{\text{no}}|}{|S|} \\ &= -\frac{4}{10}\times\log_2\frac{4}{10} - \frac{6}{10}\times\log_2\frac{6}{10} \\ &= 0.9710\end{aligned}$$

$$\begin{aligned}\text{GainRatio} &= \frac{0.5709}{0.9710} \\ &= 0.5880\end{aligned}$$

In a similar way we can compute the gain ratios Gain(S, "aquatic"), Gain(S, "aerial") and Gain(S, "has legs").

Name	Gives birth	Aquatic animal	Aerial animal	Has legs	Class label
human	yes	no	no	yes	mammal
bat	yes	no	yes	yes	bird
cat	yes	no	no	yes	mammal
shark	yes	yes	no	no	fish

Three class labels appear in this segment, namely, "bird", "fish" and "mammal". We have:

Number of examples with class label "bird"	1
Number of examples with class label "fish"	1
Number of examples with class label "mammal"	2
Total number of examples	4

Therefore we have

$$\begin{aligned}\text{Entropy}\,(S) &= \sum_{\text{for all classes "xxx"}} -p_{\text{xxx}} \log_2(p_{\text{xxx}}) \\ &= -p_{\text{bird}} \log_2(p_{\text{bird}}) - p_{\text{fish}} \log_2(p_{\text{fish}}) \\ &\quad - p_{\text{mammal}} \log_2(p_{\text{mammal}}) \\ &= -(1/4)\log_2(1/4) - (1/4)\log_2(1/4) - (2/4)\log_2(2/4) \\ &= -(1/4)\times(-2) - (1/4)\times(-2) - (2/4)\times(-1) \\ &= 1.5 \end{aligned} \tag{8.1}$$

Name	gives birth	aquatic animal	aerial animal	has legs	Class label
python	no	no	no	no	reptile
salmon	no	yes	no	no	fish
frog	no	semi	no	yes	amphibian
pigeon	no	no	yes	yes	bird
turtle	no	semi	no	yes	amphibian
salamander	no	semi	no	yes	amphibian

Four class labels appear in this segment, namely, "amphi", "bird", "fish" and "reptile". We have:

Number of examples with class label "amphi"	3
Number of examples with class label "bird"	1
Number of examples with class label "fish"	1
Number of examples with class label "reptile"	1
Total number of examples	6

Therefore, we have:

$$\begin{aligned}\text{Entropy}\,(S) &= \sum_{\text{for all classes "xxx"}} -p_{\text{xxx}} \log_2(p_{\text{xxx}}) \\ &= -p_{\text{amphi}} \log_2(p_{\text{amphi}}) - p_{\text{bird}} \log_2(p_{\text{bird}}) - p_{\text{fish}} \log_2(p_{\text{fish}}) \\ &\quad - p_{\text{reptile}} \log_2(p_{\text{reptile}}) \\ &= -(3/6)\log_2(3/6) - (1/6)\log_2(1/6) - (1/6)\log_2(1/6) \\ &\quad - (1/6)\log_2(1/6) \\ &= 1.7925\end{aligned}$$

Let O be an arbitrary observation sequence of length T. Let us consider a particular observation sequence

$$Q = (q_1, q_2, \ldots, q_T).$$

Now, given the transition matrix A and the initial probabilities Π we can calculate the probability $P(O = Q)$ as follows.

$$\begin{aligned} P(O = Q) &= P(q_1)P(q_2|q_1)P(q_3|q_2)\ldots P(q_T|q_{T-1}) \\ &= \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T} \end{aligned}$$

Here, π_{q_1} is the probability that the first state is q_1, $a_{q_1 q_2}$ is the probability of going from q_1 to q_2, and so on. We multiply these probabilities to get the probability of the whole sequence.

Consider the discrete Markov process described in Section 11.1.1. Let us compute the probability of having a bull week followed by a stagnant week followed by two bear weeks. In this case the observation sequence is

$$\begin{aligned} Q &= (\text{bull}, \text{stagnant}, \text{bear}, \text{bear}) \\ &= (S_1, S_2, S_3, S_3) \end{aligned}$$

The required probability is

$$\begin{aligned} P(O = Q) &= P(S_1)P(S_2|S_1)P(S_3|S_2)P(S_3|S_3) \\ &= \pi_1 a_{12} a_{23} a_{33} \\ &= 0.5 \times 0.075 \times 0.05 \times 0.25 \\ &= 0.00046875 \end{aligned}$$

Step 1. Obtain K observation sequences each of length T. Let q_{tk} be the observed state at time t in the k-th observation sequence.

Step 2. Let $\hat{\pi}_i$ be the estimate of the initial probability π_i. Then

$$\hat{\pi}_i = \frac{\text{number of sequences starting with } S_i}{\text{total number of sequences}}.$$

Step 3. Let $\hat{a}_{ij}$ be the estimate of a_{ij}. Then

$$\hat{a}_{ij} = \frac{\text{number of transitions from } S_i \text{ to } S_j}{\text{number of transitions from } S_i}$$

Let there be a discrete Markov process with three states S_1, S_2 and S_3. Suppose we have the following 10 observation sequences each of length 5:

$$
\begin{aligned}
O_1 &: \quad S_1\ S_2\ S_1\ S_1\ S_1 \\
O_2 &: \quad S_2\ S_1\ S_1\ S_3\ S_1 \\
O_3 &: \quad S_3\ S_1\ S_3\ S_2\ S_2 \\
O_4 &: \quad S_1\ S_3\ S_3\ S_1\ S_1 \\
O_5 &: \quad S_3\ S_2\ S_1\ S_1\ S_3 \\
O_6 &: \quad S_3\ S_1\ S_1\ S_2\ S_1 \\
O_7 &: \quad S_1\ S_1\ S_2\ S_3\ S_2 \\
O_8 &: \quad S_2\ S_3\ S_1\ S_2\ S_2 \\
O_9 &: \quad S_3\ S_2\ S_1\ S_1\ S_2 \\
O_{10} &: \quad S_1\ S_2\ S_2\ S_1\ S_1
\end{aligned}
$$

$$\hat{\pi}_1 = \frac{\text{number of sequences starting with } S_1}{\text{total number of sequences}} = \frac{4}{10}$$

$$\hat{\pi}_2 = \frac{\text{number of sequences starting with } S_2}{\text{total number of sequences}} = \frac{2}{10}$$

$$\hat{\pi}_3 = \frac{\text{number of sequences starting with } S_3}{\text{total number of sequences}} = \frac{4}{10}$$

Therefor

$$\Pi = \begin{bmatrix} 4/10 \\ 2/10 \\ 4/10 \end{bmatrix}$$

We illustrate the computation of a_{ij}'s with an example.

$$\hat{a}_{21} = \frac{\text{number of transitions from } S_2 \text{ to } S_1}{\text{number of transitions from } S_2} = \frac{6}{11}$$

$$\hat{a}_{22} = \frac{\text{number of transitions from } S_2 \text{ to } S_2}{\text{number of transitions from } S_2} = \frac{3}{11}$$

$$\hat{a}_{23} = \frac{\text{number of transitions from } S_2 \text{ to } S_3}{\text{number of transitions from } S_2} = \frac{2}{11}$$

The remaining transition probabilities can be estimated in a similar way.

- **1. What is the difference between supervised and unsupervised learning?**
- a) Supervised learning requires labeled data while unsupervised learning does not.
  b) Unsupervised learning requires labeled data while supervised learning does not.
  c) Supervised learning does not require data while unsupervised learning does.
  d) There is no difference between supervised and unsupervised learning.
- **Ans:** a

- **2. Which of the following is a type of neural network?**
- a) Decision tree
  b) Random forest
  c) Convolutional neural network
  d) Linear regression
- **Ans:** c
- **Explanation:** A convolutional neural network (CNN) is a type of neural network commonly used in image recognition tasks. Decision tree and random forest are tree-based models, while linear regression is a linear model.

- **3. What is the purpose of regularization in machine learning?**
- a) To reduce the number of features in a model
  b) To prevent overfitting and improve generalization
  c) To speed up the training process
  d) To increase the accuracy of the model
- **Ans:** b
- **Explanation:** Regularization is a technique used in machine learning to prevent overfitting of the model to the training data and improve its generalization performance. It does not reduce the number of features in a model, speed up the training process, or directly increase the accuracy of the model.

- **4. What is the difference between a validation set and a test set?**
- a) A validation set is used to tune the hyperparameters of a model, while a test set is used to evaluate its performance.
  b) A validation set is used to evaluate the performance of a model during training, while a test set is used to evaluate its performance after training.
  c) A validation set and a test set are the same thing.
  d) A validation set is not necessary in machine learning.
- **Ans:** a
- **Explanation:** A validation set is used to evaluate the performance of a model during training and tune its hyperparameters, while a test set is used to evaluate its performance after training and hyperparameter tuning. A validation set and a test set are not the same thing, and a validation set is usually necessary in machine learning to prevent overfitting.

- **5. Which of the following is an example of a classification problem?**
- a) Predicting the price of a house based on its features
  b) Predicting the weight of a person based on their height
  c) Predicting whether a customer will churn or not
  d) Predicting the age of a person based on their income
- **Ans:** c
- **Explanation:** Classification is a type of machine learning problem where the goal is to predict the class of an input, such as whether a customer will churn or not. Predicting the price of a house, weight of a person, or age of a person are regression problems.

- **6. Which of the following is an example of a clustering algorithm?**
- a) Decision tree
  b) Random forest
  c) K-means
  d) Gradient descent
- **Ans:** c
- **Explanation:** K-means is a popular clustering algorithm used in machine learning to group similar data points together. Decision tree and random forest are tree-based models, while gradient descent is an optimization algorithm used in many machine learning algorithms to minimize a loss function.

- **7. What is the purpose of feature scaling in machine learning?**
- a) To convert categorical features into numerical features
  b) To reduce the dimensionality of the feature space
  c) To standardize the range of numerical features
  d) To introduce new features into the model
- **Ans:** c
- **Explanation:** Feature scaling is a technique used to standardize the range of numerical features in a dataset. This is important because some machine learning algorithms are sensitive to the scale of the features, and feature scaling can help improve the performance and convergence of these algorithms.

- **8. What is the purpose of cross-validation in machine learning?**
- a) To evaluate the performance of a model on a held-out test set
  b) To evaluate the performance of a model on different subsets of the data
  c) To compare the performance of different models
  d) To tune the hyperparameters of a model
- **Ans:** b
- **Explanation:** Cross-validation is a technique used in machine learning to evaluate the performance of a model on different subsets of the data, in order to assess its generalization performance and detect overfitting. It does not involve a held-out test set or hyperparameter tuning, but can be used to compare the performance of different models.

- **9. Which of the following is an example of a dimensionality reduction technique?**
- a) Principal component analysis (PCA)
  b) Support vector machine (SVM)
  c) K-nearest neighbors (KNN)
  d) AdaBoost
- **Ans:** a
- **Explanation:** Principal component analysis (PCA) is a dimensionality reduction technique used in machine learning to reduce the number of features in a dataset while retaining as much information as possible. Support vector machine (SVM), K-nearest neighbors (KNN), and AdaBoost are machine learning algorithms that do not involve dimensionality reduction.

- **10. What is the purpose of the confusion matrix in machine learning?**
- a) To visualize the distribution of the data in a dataset
  b) To compare the performance of different models
  c) To evaluate the performance of a classification model
  d) To evaluate the performance of a regression model
- **Ans:** c
- **Explanation:** A confusion matrix is a table used in machine learning to evaluate the performance of a classification model by comparing its predicted labels to the true labels in the test set. It can be used to calculate metrics such as accuracy, precision, recall, and F1 score.

- **11. Which of the following is a measure of model complexity?**
- a) Mean squared error (MSE)
  b) R-squared (R2)
  c) Akaike information criterion (AIC)
  d) Bayesian information criterion (BIC)
- **Ans:** c
- **Explanation:** The Akaike information criterion (AIC) is a measure of model complexity used in machine learning and statistics to compare the performance of different models. It takes into account both the goodness of fit and the number of parameters in the model, and penalizes models with more parameters.

- **12. What is the purpose of data augmentation in machine learning?**
- a) To increase the size of a dataset
  b) To reduce the size of a dataset
  c) To improve the quality of a dataset
  d) To improve the performance of a model
- **Ans:** a
- **Explanation:** Data augmentation is a technique used in machine learning to increase the size of a dataset by creating new examples from the existing ones, typically by applying random transformations such as rotations, translations, or flips. This can help improve the performance of a model by providing more training data and reducing overfitting.

- **13. Which of the following is an example of a supervised learning problem?**
- a) Image classification
  b) Market segmentation
  c) Fraud detection
  d) Social network analysis
- **Ans:** a
- **Explanation:** Image classification is an example of a supervised learning problem, where the goal is to learn a mapping from input images to output labels (e.g., object categories). Market segmentation, fraud detection, and social network analysis are examples of unsupervised or semi-supervised learning problems.

- **14. Which of the following is an example of an unsupervised learning problem?**
- a) Predicting the stock market
  b) Recommending products to users
  c) Spam filtering
  d) Sentiment analysis
- **Ans:** b
- **Explanation:** Recommending products to users is an example of an unsupervised learning problem, where the goal is to learn a model that can predict a user's preferences or interests based on their past behavior or other data. Predicting the stock market, spam filtering, and sentiment analysis are examples of supervised learning problems.

- **15. What is the purpose of regularization in machine learning?**
- a) To prevent overfitting
  b) To increase the accuracy of the model
  c) To reduce the variance of the model
  d) To reduce the bias of the model
- **Ans:** a
- **Explanation:** Regularization is a technique used in machine learning to prevent overfitting by adding a penalty term to the loss function that discourages the model from learning complex or noisy patterns in the training data. This can help improve the generalization performance of the model on unseen data.

- **16. Which of the following is an example of a non-parametric machine learning algorithm?**
- a) Linear regression
  b) Logistic regression
  c) Decision tree
  d) Support vector machine
- **Ans:** c
- **Explanation:** A decision tree is an example of a non-parametric machine learning algorithm, which does not make any assumptions about the underlying distribution of the data or the functional form of the model. Linear regression, logistic regression, and support vector machine are examples of parametric machine learning algorithms.

- **17. Which of the following is an example of a deep learning architecture?**
- a) K-nearest neighbors (KNN)
  b) Random forest
  c) Convolutional neural network (CNN)
  d) Gradient boosting machine (GBM)
- **Ans:** c
- **Explanation:** A convolutional neural network (CNN) is an example of a deep learning architecture, which consists of multiple layers of non-linear transformations that can learn hierarchical representations of the input data. K-nearest neighbors (KNN), random forest, and gradient boosting machine (GBM) are examples of classical machine learning algorithms that do not involve deep learning.

- **18. Which of the following is an example of a semi-supervised learning problem?**
- a) Image classification
  b) Object detection
  c) Text clustering
  d) Speech recognition
- **Ans:** c
- **Explanation:** Text clustering is an example of a semi-supervised learning problem, where some of the data points are labeled (e.g., with their topics), but many others are unlabeled. The goal is to learn a model that can group similar documents together based on their content, using both the labeled and unlabeled data.

- **19. Which of the following is a common activation function used in deep learning?**
- a) Sigmoid
  b) Linear
  c) Exponential
  d) Quadratic
- **Ans:** a
- **Explanation:** The sigmoid function is a common activation function used in deep learning, which maps the output of a neuron to a value between 0 and 1, representing its activation level. Other common activation functions in deep learning include ReLU (rectified linear unit), tanh (hyperbolic tangent), and softmax (used for multi-class classification).

- **20. Which of the following is a hyperparameter in machine learning?**
- a) Learning rate
  b) Training data
  c) Test data
  d) Validation set
- **Ans:** a
- **Explanation:** A hyperparameter is a parameter that is set before the training process begins and cannot be learned directly from the data. Examples of hyperparameters include the learning rate, which determines the step size taken during gradient descent optimization, and the number of hidden units in a neural network, which controls its capacity and complexity.

- **21. Which of the following is a common evaluation metric for binary classification?**
- a) Accuracy
  b) F1 score
  c) Mean squared error (MSE)
  d) Area under the ROC curve (AUC)
- **Ans:** d
- **Explanation:** The area under the ROC curve (AUC) is a common evaluation metric for binary classification, which measures the performance of a classifier at different threshold values for the predicted probabilities. Other common metrics include accuracy, precision, recall, and F1 score, which are based on the confusion matrix of true positives, false positives, true negatives, and false negatives.

- **23. Which of the following is a common regularization technique for linear regression?**
- a) L1 regularization (Lasso)
  b) L2 regularization (Ridge)
  c) Dropout
  d) Batch normalization
- **Ans:** b
- **Explanation:** L2 regularization, also known as Ridge regression, is a common technique for linear regression that adds a penalty term to the loss function based on the squared magnitude of the model weights, which helps to prevent overfitting. L1 regularization (Lasso) is another popular technique that uses a penalty term based on the absolute magnitude of the weights, and dropout and batch normalization are techniques used in neural networks to regularize the activations and gradients.

- **24. Which of the following is an example of a clustering algorithm?**
- a) Linear regression
  b) Logistic regression
  c) K-means
  d) Support vector machine
- **Ans:** c
- **Explanation:** K-means is a clustering algorithm that partitions a set of data points into K clusters based on their similarity, using an iterative algorithm that updates the cluster centroids to minimize the sum of squared distances from the data points to their assigned cluster. Linear regression, logistic regression, and support vector machine are examples of supervised learning algorithms that are not used for clustering.

- **25. Which of the following is a common approach to reducing dimensionality in machine learning?**
- a) Feature selection
  b) Feature extraction
  c) Feature scaling
  d) Feature engineering
- **Ans:** b
- **Explanation:** Feature extraction is a common approach to reducing dimensionality in machine learning, which involves transforming the original features into a new set of features that capture the relevant information in a more compact and informative way. Examples of feature extraction techniques include principal component analysis (PCA), which projects the data onto a lower-dimensional subspace that captures the most variance, and autoencoders, which learn a compressed representation of the data by encoding and decoding it through a neural network. Feature selection, feature scaling, and feature engineering are other important techniques for preprocessing the data, but they do not necessarily reduce the dimensionality of the feature space.

- **26. Which of the following is a common approach to ensemble learning?**
- a) Bagging
  b) Boosting
  c) Stacking
  d) All of the above
- **Ans:** d
- **Explanation:** Ensemble learning is a powerful technique for improving the performance and robustness of machine learning models by combining multiple base models into a single prediction. There are several approaches to ensemble learning, including bagging, which involves training multiple models on different subsets of the training data and aggregating their predictions, boosting, which involves sequentially training models on the misclassified samples and weighting their predictions, and stacking, which involves training a meta-model that combines the outputs of multiple base models as input features.

- **1. What is Scikit-learn?**
- A) A machine learning library in Python
  B) A data visualization library in Python
  C) A natural language processing library in Python
  D) A web development framework in Python
- **Ans:** A
- **Explanation:** Scikit-learn is an open-source machine learning library in Python that provides a range of tools for supervised and unsupervised learning tasks, including classification, regression, clustering, and dimensionality reduction, among others.

- **2. What is the purpose of the fit() method in Scikit-learn?**
- A) To train a model using a given dataset
  B) To make predictions using a trained model
  C) To evaluate the performance of a model
  D) To visualize the data using a plot
- **Ans:** A
- **Explanation:** The fit() method is used to train a model using a given dataset. It fits the model parameters to the data, adjusting them to minimize the error between the predicted output and the actual output.

- **3. Which of the following is an example of a supervised learning algorithm?**
- A) K-means clustering
  B) Decision tree
  C) Principal component analysis (PCA)
  D) Apriori algorithm
- **Ans:** B
- **Explanation:** Decision tree is an example of a supervised learning algorithm, where the model is trained on labeled data to make predictions on new, unseen data.

- **4. Which of the following is NOT a classification metric used in Scikit-learn?**
- A) Precision
  B) Recall
  C) F1-score
  D) R-squared
- **Ans:** D
- **Explanation:** R-squared is a regression metric used to measure the goodness of fit of a model, while the other options are classification metrics used to evaluate the performance of a classification model.

- **5. Which of the following is a clustering algorithm in Scikit-learn?**
- A) Random forest
  B) K-means
  C) Support vector machines (SVM)
  D) Gradient boosting
- **Ans:** B
- **Explanation:** K-means is a clustering algorithm in Scikit-learn that groups similar data points together based on their distance from the cluster centroids.

- **6. Which of the following is an example of a dimensionality reduction algorithm in Scikit-learn?**
- A) Linear regression
  B) K-nearest neighbors (KNN)
  C) Principal component analysis (PCA)
  D) Naive Bayes
- **Ans:** C
- **Explanation:** PCA is a dimensionality reduction algorithm that transforms high-dimensional data into a lower-dimensional representation while preserving as much of the original variance as possible.

- **7. What is the purpose of the predict() method in Scikit-learn?**
- A) To train a model using a given dataset
  B) To make predictions using a trained model
  C) To evaluate the performance of a model
  D) To visualize the data using a plot
- **Ans:** B
- **Explanation:** The predict() method is used to make predictions on new, unseen data using a trained model.

- **8. Which of the following is NOT a preprocessing step in Scikit-learn?**
- A) Scaling
  B) Imputation
  C) Encoding
  D) Regularization
- **Ans:** D
- **Explanation:** Regularization is a model parameter tuning technique used to prevent overfitting in machine learning models, while the other options are preprocessing steps used to prepare the data for modeling.

- **9. Which of the following is an ensemble learning algorithm in Scikit-learn?**
- A) K-means clustering
  B) Decision tree
  C) Random forest
  D) Linear regression
- **Ans:** C
- **Explanation:** Random forest is an ensemble learning algorithm in Scikit-learn that combines multiple decision trees to improve the accuracy and robustness of the model.

- **10. What is the purpose of the score() method in Scikit-learn?**
- A) To train a model using a given dataset
  B) To make predictions using a trained model
  C) To evaluate the performance of a model
  D) To visualize the data using a plot
- **Ans:** C
- **Explanation:** The score() method is used to evaluate the performance of a trained model using a given metric, such as accuracy or mean squared error.

- **11. Which of the following is an example of a regression algorithm in Scikit-learn?**
- A) K-means clustering
  B) Decision tree
  C) Linear regression
  D) Support vector machines (SVM)
- **Ans:** C
- **Explanation:** Linear regression is an example of a regression algorithm in Scikit-learn, where the model is trained on labeled data to predict a continuous output variable.

- **12. What is cross-validation in Scikit-learn?**
- A) A method for evaluating the performance of a model
  B) A method for preprocessing the data
  C) A method for selecting the best features
  D) A method for tuning the hyperparameters of a model
- **Ans:** A
- **Explanation:** Cross-validation is a method for evaluating the performance of a model by splitting the data into multiple folds, training the model on one fold and evaluating it on the remaining folds, and repeating this process for each fold.

- **14. What is the purpose of the transform() method in Scikit-learn?**
- A) To train a model using a given dataset
  B) To make predictions using a trained model
  C) To evaluate the performance of a model
  D) To preprocess the data for modeling
- **Ans:** D
- **Explanation:** The transform() method is used to preprocess the data for modeling, such as scaling or encoding the features, before training a model.

- **15. Which of the following is a metric used for clustering evaluation in Scikit-learn?**
- A) Precision
  B) Recall
  C) F1-score
  D) Silhouette score
- **Ans:** D
- **Explanation:** Silhouette score is a metric used for clustering evaluation in Scikit-learn, which measures the similarity of data points within a cluster and the dissimilarity between different clusters.

- **16. Which of the following is an example of a semi-supervised learning algorithm?**
- A) Decision tree
  B) K-means clustering
  C) Support vector machines (SVM)
  D) Label propagation
- **Ans:** D
- **Explanation:** Label propagation is an example of a semi-supervised learning algorithm that uses a small amount of labeled data and a larger amount of unlabeled data to make predictions on new, unseen data.

- **1. What is TensorFlow?**
  A) A machine learning library
  B) A programming language
  C) A deep learning framework
  D) A database management system
- **Ans:** A
- **Explanation:** TensorFlow is an open-source machine learning library developed by Google Brain Team. It is widely used for numerical computations and building neural networks.

- **2. What is a tensor in TensorFlow?**
- A) A type of data structure
  B) A machine learning model
  C) A database management system
  D) A programming language
- **Ans:** A
- **Explanation:** A tensor is a type of data structure used in TensorFlow for representing multi-dimensional arrays or matrices.

- **3. What is the default data type of TensorFlow tensors?**
- A) int64
  B) float32
  C) double
  D) int32
- **Ans:** B
- **Explanation:** The default data type of TensorFlow tensors is float32, which is a 32-bit floating-point number.

- **4. Which of the following is NOT a valid TensorFlow data type?**
- A) int32
  B) bool
  C) float16
  D) char
- **Ans:** D
- **Explanation:** Char is not a valid TensorFlow data type. The valid data types are int32, bool, float16, float32, float64, and complex64.

- **5. What is a placeholder in TensorFlow?**
- A) A variable that holds the output of a neural network
  B) A variable that holds the input data for a neural network
  C) A variable that holds the weights of a neural network
  D) A variable that holds the bias of a neural network
- **Ans:** B
- **Explanation:** A placeholder is a variable in TensorFlow that holds the input data for a neural network. It is used to feed data into the network during training.

- **6. What is a variable in TensorFlow?**
- A) A fixed value that is used in a neural network
  B) A data structure that holds the input data for a neural network
  C) A data structure that holds the weights and biases of a neural network
  D) A fixed value that is used to compute the output of a neural network
- **Ans:** C
- **Explanation:** A variable is a data structure in TensorFlow that holds the weights and biases of a neural network. It is updated during training to improve the performance of the network.

- **16. What is transfer learning in TensorFlow?**
  A) A technique for initializing the weights and biases of a neural network
  B) A technique for updating the weights and biases of a neural network
  C) A technique for measuring the difference between the predicted output and the actual output
  D) A technique for reusing pre-trained neural network models
- **Ans:** D
- **Explanation:** Transfer learning is a technique in TensorFlow for reusing pre-trained neural network models to solve a new task. It involves using the learned features of the pre-trained model as a starting point for
  training a new model on a different dataset.

- **21. What is the purpose of a confusion matrix in TensorFlow?**
- A) To measure the accuracy of a classification model
  B) To measure the recall of a classification model
  C) To measure the precision of a classification model
  D) To visualize the performance of a classification model
- **Ans:** D
- **Explanation:** A confusion matrix is a visualization tool in TensorFlow used to display the performance of a classification model. It shows the number of correct and incorrect predictions for each class in a tabular format.

- **What is precision in TensorFlow?**
- A) The ratio of true positives to the sum of true positives and false positives
  B) The ratio of true positives to the sum of true positives and false negatives
  C) The ratio of true positives to the total number of positive examples
  D) The ratio of true negatives to the total number of negative examples
- **Ans:** A
- **Explanation:** Precision in TensorFlow is the ratio of true positives to the sum of true positives and false positives. It measures the proportion of positive predictions that are actually correct.

- **What is recall in TensorFlow?**
  A) The ratio of true positives to the sum of true positives and false positives
  B) The ratio of true positives to the sum of true positives and false negatives
  C) The ratio of true positives to the total number of positive examples
  D) The ratio of true negatives to the total number of negative examples
- **Ans:** B
- **Explanation:** Recall in TensorFlow is the ratio of true positives to the sum of true positives and false negatives. It measures the proportion of actual positive examples that are correctly identified by the model.

- **What is F1 score in TensorFlow?**
- A) The harmonic mean of precision and recall
  B) The arithmetic mean of precision and recall
  C) The maximum of precision and recall
  D) The minimum of precision and recall
- **Ans:** A
- **Explanation:** F1 score in TensorFlow is the harmonic mean of precision and recall. It provides a balanced measure of the model's accuracy by taking into account both precision and recall.

- **What is transfer learning in TensorFlow?**
- A) A technique for training a model on a small dataset and then fine-tuning it on a larger dataset
  B) A technique for training a model on a large dataset and then fine-tuning it on a small dataset
  C) A technique for training a model on a dataset with one set of labels and then using it to classify a new set of labels
  D) A technique for training a model on a dataset with multiple tasks and then using it to perform a new task
- **Ans:** A
- **Explanation:** Transfer learning is a technique in TensorFlow for training a model on a small dataset and then fine-tuning it on a larger dataset. It involves using a pre-trained model as a starting point and then adapting it to the new dataset.

- **What is a pre-trained model in TensorFlow?**
- A) A model that has been trained on a large dataset and can be used as a starting point for a new task
  B) A model that has been trained on a small dataset and is ready to be deployed
  C) A model that has been trained on a large dataset and is ready to be deployed
  D) A model that has not been trained yet and needs to be trained from scratch
  Ans: A
- **Explanation:** A pre-trained model in TensorFlow is a model that has been trained on a large dataset and can be used as a starting point for a new task. It is often used for transfer learning.


THANKS!

Cheat Sheet – Regularization in ML

What is Regularization in ML?

- Regularization is an approach to address over-fitting in ML.
- Overfitted model fails to generalize estimations on test data
- When the underlying model to be learned is low bias/high variance, or when we have small amount of data, the estimated model is prone to over-fitting.
- Regularization reduces the variance of the model

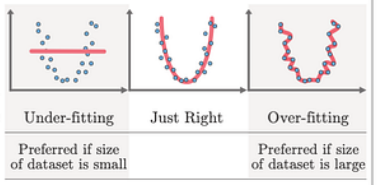


Figure 1. Overfitting

Types of Regularization:

1. Modify the loss function:

- **L2 Regularization:** Prevents the weights from getting too large (defined by L2 norm). Larger the weights, more complex the model is, more chances of overfitting.

$$loss = error(y, \hat{y}) + \lambda \sum_j \beta_j^2 \quad \lambda \geq 0,\ \lambda \propto model\ bias,\quad \lambda \propto \frac{1}{model\ variance}$$

- **L1 Regularization:** Prevents the weights from getting too large (defined by L1 norm). Larger the weights, more complex the model is, more chances of overfitting. L1 regularization introduces sparsity in the weights. It forces more weights to be zero, than reducing the the average magnitude of all weights

$$loss = error(y, \hat{y}) + \lambda \sum_j |\beta_j| \quad \lambda \geq 0,\ \lambda \propto model\ bias,\quad \lambda \propto \frac{1}{model\ variance}$$

- **Entropy:** Used for the models that output probability. Forces the probability distribution towards uniform distribution.

$$loss = error(p, \hat{p}) - \lambda \sum_i \hat{p}_i log(\hat{p}_i) \quad \lambda \geq 0,\ \lambda \propto model\ bias,\quad \lambda \propto \frac{1}{model\ variance}$$

2. Modify data sampling:

- **Data augmentation:** Create more data from available data by randomly cropping, dilating, rotating, adding small amount of noise etc.
- **K-fold Cross-validation:** Divide the data into k groups. Train on (k-1) groups and test on 1 group. Try all k possible combinations.

3. Change training approach:

- **Injecting noise:** Add random noise to the weights when they are being learned. It pushes the model to be relatively insensitive to small variations in the weights, hence regularization
- **Dropout:** Generally used for neural networks. Connections between consecutive layers are randomly dropped based on a dropout-ratio and the remaining network is trained in the current iteration. In the next iteration, another set of random connections are dropped.

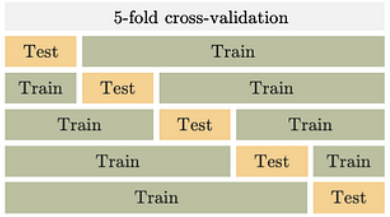


Figure 2. K-fold CV

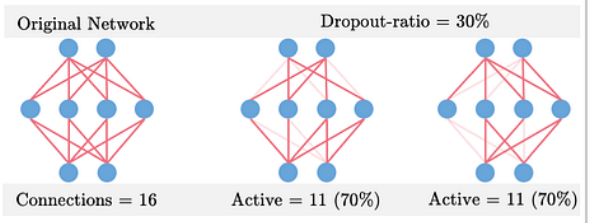
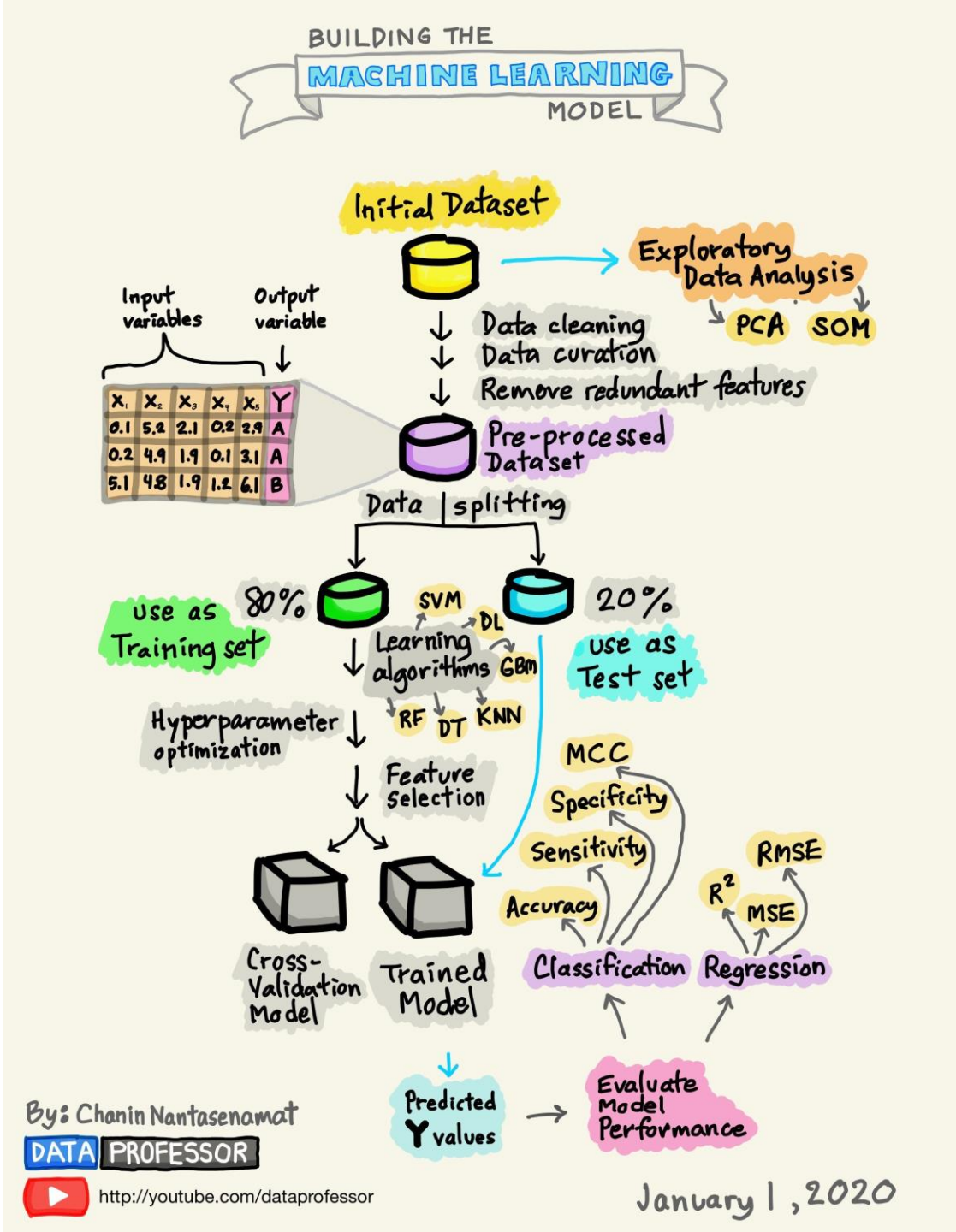


Figure 3. Drop-out

Source: https://www.cheatsheets.aqeel-anwar.com

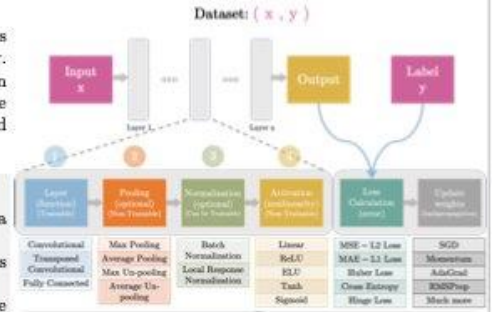
Dr.Arun Anoop M



Cheat Sheet – Convolutional Neural Network

Convolutional Neural Network:

The data gets into the CNN through the input layer and passes through various hidden layers before getting to the output layer. The output of the network is compared to the actual labels in terms of loss or error. The partial derivatives of this loss w.r.t the trainable weights are calculated, and the weights are updated through one of the various methods using backpropagation.



CNN Template:

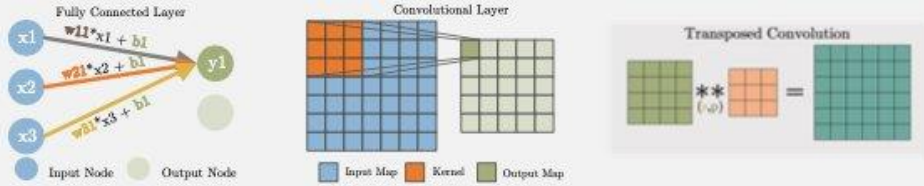
Most of the commonly used hidden layers (not all) follow a pattern

1. **Layer function:** Basic transforming function such as convolutional or fully connected layer.

a. **Fully Connected:** Linear functions between the input and the

a. **Convolutional Layers:** These layers are applied to 2D (3D) input feature maps. The trainable weights are a 2D (3D) kernel/filter that moves across the input feature map, generating dot products with the overlapping region of the input feature map.

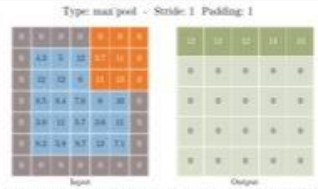
b. **Transposed Convolutional (DeConvolutional) Layer:** Usually used to increase the size of the output feature map (Upsampling) The idea behind the transposed convolutional layer is to undo (not exactly) the convolutional layer



2. **Pooling:** Non-trainable layer to change the size of the feature map

a. **Max/Average Pooling:** Decrease the spatial size of the input layer based on selecting the maximum/average value in receptive field defined by the kernel

b. **UnPooling:** A non-trainable layer used to increase the spatial size of the input layer based on placing the input pixel at a certain index in the receptive field of the output defined by the kernel.



3. **Normalization:** Usually used just before the activation functions to limit the unbounded activation from increasing the output layer values too high

a. **Local Response Normalization LRN:** A **non-trainable layer** that square-normalizes the pixel values in a feature map within a local neighborhood.

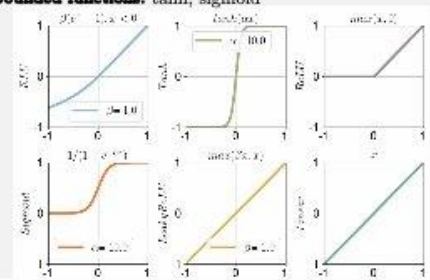
b. **Batch Normalization:** A trainable approach to normalizing the data by learning scale and shift variable during training.

3. **Activation:** Introduce non-linearity so CNN can efficiently map non-linear complex mapping.

a. **Non-parametric/Static functions:** Linear, ReLU

b. **Parametric functions:** ELU, tanh, sigmoid, Leaky ReLU

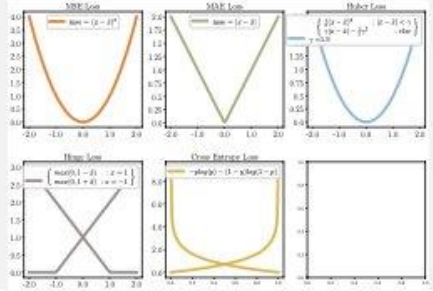
c. **Bounded functions:** tanh, sigmoid



5. **Loss function:** Quantifies how far off the CNN prediction is from the actual labels.

a. Regression Loss Functions: MAE, MSE, Huber loss

b. Classification Loss Functions: Cross entropy, Hinge loss



Source: https://www.cheatsheets.aqeel-anwar.com Tutorial: Click here

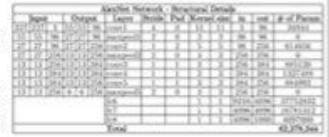
Cheat Sheet – Famous CNNs

AlexNet – 2012

Why: AlexNet was born out of the need to improve the results of the ImageNet challenge.

What: The network consists of 5 Convolutional (CONV) layers and 3 Fully Connected (FC) layers. The activation used is the Rectified Linear Unit (ReLU).

How: Data augmentation is carried out to reduce over-fitting, Uses Local response localization.

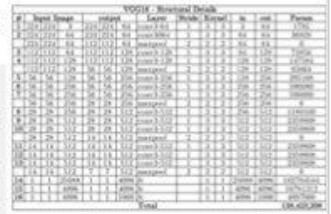


VGGNet – 2014

Why: VGGNet was born out of the need to reduce the # of parameters in the CONV layers and improve on training time

What: There are multiple variants of VGGNet (VGG16, VGG19, etc.)

How: The important point to note here is that all the conv kernels are of size 3x3 and maxpool kernels are of size 2x2 with a stride of two.

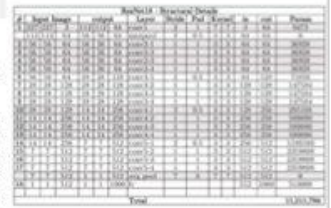


ResNet – 2015

Why: Neural Networks are notorious for not being able to find a simpler mapping when it exists. ResNet solves that.

What: There are multiple versions of ResNetXX architectures where 'XX' denotes the number of layers. The most used ones are ResNet50 and ResNet101. Since the vanishing gradient problem was taken care of (more about it in the How part), CNN started to get deeper and deeper

How: ResNet architecture makes use of shortcut connections do solve the vanishing gradient problem. The basic building block of ResNet is a Residual block that is repeated throughout the network.



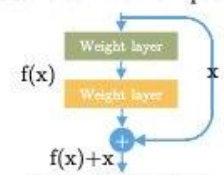


Figure 1 ResNet Block

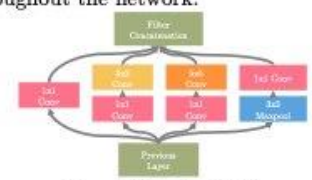


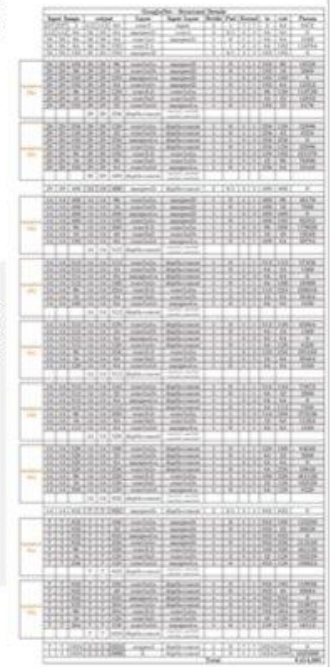
Figure 2 Inception Block

Inception – 2014

Why: Lager kernels are preferred for more global features, on the other hand, smaller kernels provide good results in detecting area-specific features. For effective recognition of such a variable-sized feature, we need kernels of different sizes. That is what Inception does.

What: The Inception network architecture consists of several inception modules of the following structure. Each inception module consists of four operations in parallel, 1x1 conv layer, 3x3 conv layer, 5x5 conv layer, max pooling

How: Inception increases the network space from which the best network is to be chosen via training. Each inception module can capture salient features at different levels.



Network	Year	Salient Feature	top5 accuracy	Parameters	FLOP
AlexNet	2012	Deeper	84.70%	62M	1.5B
VGGNet	2014	Fixed-size kernels	92.30%	138M	19.6B
Inception	2014	Wider - Parallel kernels	93.30%	6.4M	2B
ResNet-152	2015	Shortcut connections	95.51%	60.3M	11B

Source: https://www.cheatsheets.aqeel-anwar.com Tutorial: Click here

Cheat Sheet – Regression Analysis

What is Regression Analysis?

Fitting a function f(.) to datapoints $y_i=f(x_i)$ under some error function. Based on the estimated function and error, we have the following types of regression

1. Linear Regression:

Fits a line minimizing the sum of mean-squared error for each datapoint.

$$min_\beta \sum_i \|y_i - f_\beta^{linear}(x_i)\|^2$$

$$f_\beta^{linear}(x_i) = \beta_0 + \beta_1 x_i$$

2. Polynomial Regression:

Fits a polynomial of order k (k+1 unknowns) minimizing the sum of mean-squared error for each datapoint.

$$min_\beta \sum_{i=0}^{m} \|y_i - f_\beta^{poly}(x_i)\|^2$$

$$f_\beta^{poly}(x_i) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_k x_i^k$$

3. Bayesian Regression:

For each datapoint, fits a gaussian distribution by minimizing the mean-squared error. As the number of data points x_i increases, it converges to point estimates i.e. $n \to \infty, \sigma^2 \to 0$

$$min_\beta \sum_i \|y_i - \mathcal{N}(f_\beta(x_i), \sigma^2)\|^2$$

$$f_\beta(x_i) = f_\beta^{poly}(x_i) \text{ or } f_\beta^{linear}(x_i)$$

$\mathcal{N}(\mu, \sigma^2) \to$ Gaussian with mean μ and variance σ^2

4. Ridge Regression:

Can fit either a line, or polynomial minimizing the sum of mean-squared error for each datapoint and the weighted L2 norm of the function parameters beta.

$$min_\beta \sum_{i=0}^{m} \|y_i - f_\beta(x_i)\|^2 + \sum_{j=0}^{k} \beta_j^2$$

$$f_\beta(x_i) = f_\beta^{poly}(x_i) \text{ or } f_\beta^{linear}(x_i)$$

5. LASSO Regression:

Can fit either a line, or polynomial minimizing the the sum of mean-squared error for each datapoint and the weighted L1 norm of the function parameters beta.

$$min_\beta \sum_{i=0}^{m} \|y_i - f_\beta(x_i)\|^2 + \sum_{j=0}^{k} |\beta_j|$$

$$f_\beta(x_i) = f_\beta^{poly}(x_i) \text{ or } f_\beta^{linear}(x_i)$$

6. Logistic Regression:

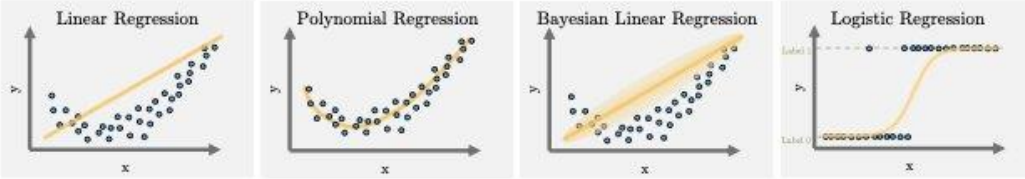
Can fit either a line, or polynomial with sigmoid activation minimizing the binary cross-entropy loss for each datapoint. The labels y are binary class labels.

$$min_\beta \sum_i -y_i log(\sigma(f_\beta(x_i))) - (1 - y_i) log(1 - \sigma(f_\beta(x_i)))$$

$$f_\beta(x_i) = f_\beta^{poly}(x_i) \text{ or } f_\beta^{linear}(x_i)$$

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

Visual Representation:



Summary:

	What does it fit?	Estimated function	Error Function
Linear	A line in n dimensions	$f_\beta^{linear}(x_i) = \beta_0 + \beta_1 x_i$	$\sum_{i=0}^{m} \|y_i - f_\beta(x_i)\|^2$
Polynomial	A polynomial of order k	$f_\beta^{poly}(x_i) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots$	$\sum_{i=0}^{m} \|y_i - f_\beta(x_i)\|^2$
Bayesian Linear	Gaussian distribution for each point	$\mathcal{N}(f_\beta(x_i), \sigma^2)$	$\sum_i \|y_i - \mathcal{N}(f_\beta(x_i), \sigma^2)\|^2$
Ridge	Linear/polynomial	$f_\beta^{poly}(x_i)$ or $f_\beta^{linear}(x_i)$	$\sum_{i=0}^{m} \|y_i - f_\beta(x_i)\|^2 + \sum_{j=0}^{k} \beta_j^2$
LASSO	Linear/polynomial	$f_\beta^{poly}(x_i)$ or $f_\beta^{linear}(x_i)$	$\sum_{i=0}^{m} \|y_i - f_\beta(x_i)\|^2 + \sum_{j=0}^{k} \|\beta_j\|$
Logistic	Linear/polynomial with sigmoid	$\sigma(f_\beta(x_i))$	$min_\beta \sum_i -y_i log(\sigma(f_\beta(x_i))) - (1 - y_i) log(1 - \sigma(f_\beta(x_i)))$

Source: https://www.cheatsheets.aqeel-anwar.com Tutorial: Click here

Cheat Sheet – Regularization in ML

What is Regularization in ML?

- Regularization is an approach to address over-fitting in ML.
- Overfitted model fails to generalize estimations on test data
- When the underlying model to be learned is low bias/high variance, or when we have small amount of data, the estimated model is prone to over-fitting.
- Regularization reduces the variance of the model

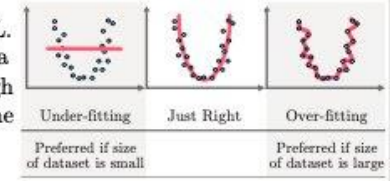


Figure 1. Overfitting

Types of Regularization:

1. Modify the loss function:

- **L2 Regularization:** Prevents the weights from getting too large (defined by L2 norm). Larger the weights, more complex the model is, more chances of overfitting.

$$loss = error(y, \hat{y}) + \lambda \sum_j \beta_j^2 \quad \lambda \geq 0,\ \lambda \propto model\ bias,\ \lambda \propto \frac{1}{model\ variance}$$

- **L1 Regularization:** Prevents the weights from getting too large (defined by L1 norm). Larger the weights, more complex the model is, more chances of overfitting. L1 regularization introduces sparsity in the weights. It forces more weights to be zero, than reducing the the average magnitude of all weights

$$loss = error(y, \hat{y}) + \lambda \sum_j |\beta_j| \quad \lambda \geq 0,\ \lambda \propto model\ bias,\ \lambda \propto \frac{1}{model\ variance}$$

- **Entropy:** Used for the models that output probability. Forces the probability distribution towards uniform distribution.

$$loss = error(p, \hat{p}) - \lambda \sum_i \hat{p}_i log(\hat{p}_i) \quad \lambda \geq 0,\ \lambda \propto model\ bias,\ \lambda \propto \frac{1}{model\ variance}$$

2. Modify data sampling:

- **Data augmentation:** Create more data from available data by randomly cropping, dilating, rotating, adding small amount of noise etc.
- **K-fold Cross-validation:** Divide the data into k groups. Train on (k-1) groups and test on 1 group. Try all k possible combinations.

3. Change training approach:

- **Injecting noise:** Add random noise to the weights when they are being learned. It pushes the model to be relatively insensitive to small variations in the weights, hence regularization
- **Dropout:** Generally used for neural networks. Connections between consecutive layers are randomly dropped based on a dropout-ratio and the remaining network is trained in the current iteration. In the next iteration, another set of random connections are dropped.

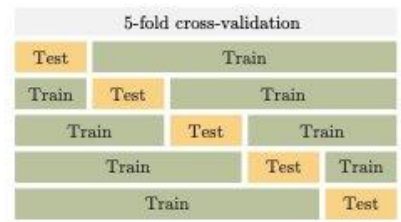


Figure 2. K-fold CV

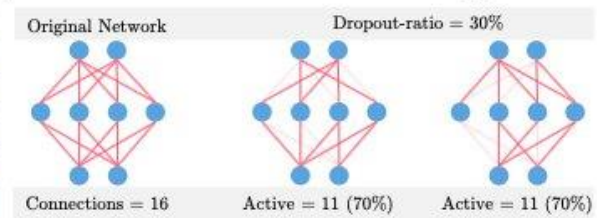


Figure 3. Drop-out

Source: https://www.cheatsheets.aqeel-anwar.com Tutorial: Click here

Cheat Sheet – PCA Dimensionality Reduction

What is PCA?

- Based on the dataset find a new set of orthogonal feature vectors in such a way that the data spread is maximum in the direction of the feature vector (or dimension)
- Rates the feature vector in the decreasing order of data spread (or variance)
- The datapoints have maximum variance in the first feature vector, and minimum variance in the last feature vector
- The variance of the datapoints in the direction of feature vector can be termed as a measure of information in that direction.

Steps

1. Standardize the datapoints $X_{new} = \frac{X - mean(X)}{std(X)}$
2. Find the covariance matrix from the given datapoints $C[i,j] = cov(x_i, x_j)$
3. Carry out eigen-value decomposition of the covariance matrix $C = V\Sigma V^{-1}$
4. Sort the eigenvalues and eigenvectors $\Sigma_{sort} = sort(\Sigma)$ $V_{sort} = sort(V, \Sigma_{sort})$

Dimensionality Reduction with PCA

- Keep the first m out of n feature vectors rated by PCA. These m vectors will be the best m vectors preserving the maximum information that could have been preserved with m vectors on the given dataset

Steps:

1. Carry out steps 1-4 from above
2. Keep first m feature vectors from the sorted eigenvector matrix $V_{reduced} = V[:, 0:m]$
3. Transform the data for the new basis (feature vectors) $X_{reduced} = X_{new} \times V_{reduced}$
4. The importance of the feature vector is proportional to the magnitude of the eigen value

Figure 1

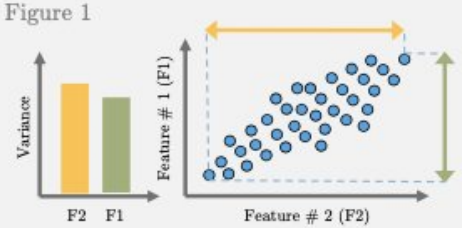


Figure 2

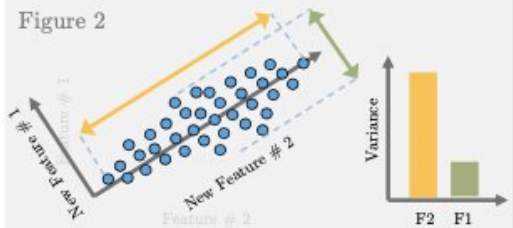


Figure 3

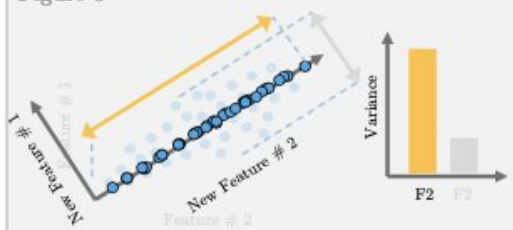


Figure 1: Datapoints with feature vectors as x and y-axis
Figure 2: The cartesian coordinate system is rotated to maximize the standard deviation along any one axis (new feature # 2)
Figure 3: Remove the feature vector with minimum standard deviation of datapoints (new feature # 1) and project the data on new feature # 2

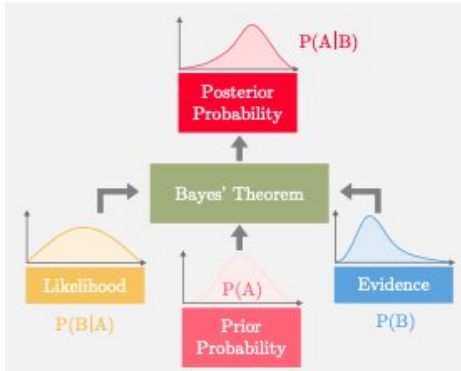
Cheat Sheet – Bayes Theorem and Classifier

What is Bayes' Theorem?

- Describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

$$P(A|B) = \frac{P(B|A)(likelihood) \times P(A)(prior)}{P(B)(evidence)}$$

- How the probability of an event changes when we have knowledge of another event
  P(A) ⟶ P(A|B)
  Usually, a better estimate than P(A)



Example

- Probability of fire P(F) = 1%
- Probability of smoke P(S) = 10%
- Prob of smoke given there is a fire P(S|F) = 90%
- What is the probability that there is a fire given we see a smoke P(F|S)?

$$P(F|S) = \frac{P(S|F) \times P(F)}{P(S)} = \frac{0.9 \times 0.01}{0.1} = 9\%$$

Maximum Aposteriori Probability (MAP) Estimation

The MAP estimate of the random variable y, given that we have observed iid (x_1, x_2, x_3, …), is given by. We try to accommodate our prior knowledge when estimating.

$$\hat{y}_{MAP} = argmax_y \; P(y) \prod_i P(x_i|y)$$

y that maximizes the product of prior and likelihood

Maximum Likelihood Estimation (MLE)

The MAP estimate of the random variable y, given that we have observed iid (x_1, x_2, x_3, …), is given by. We assume we don't have any prior knowledge of the quantity being estimated.

$$\hat{y}_{MLE} = argmax_y \prod_i P(x_i|y)$$

y that maximizes only the likelihood

MLE is a special case of MAP where our prior is uniform (all values are equally likely)

Naïve Bayes' Classifier (Instantiation of MAP as classifier)

Suppose we have two classes, $y=y_1$ and $y=y_2$. Say we have more than one evidence/features (x_1, x_2, x_3, …), using Bayes' theorem

$$P(y|x_1, x_2, x_3, \ldots) = \frac{P(x_1, x_2, x_3, \ldots|y) \times P(y)}{P(x_1, x_2, x_3, \ldots)}$$

Naïve Bayes' theorem assumes the features (x_1, x_2, …) are i.i.d. i.e $P(x_1, x_2, x_3, \ldots|y) = \prod_i P(x_i|y)$

$$P(y|x_1, x_2, x_3, \ldots) = \prod_i P(x_i|y) \frac{P(y)}{P(x_1, x_2, x_3, \ldots)}$$

$$\hat{y} = y_1 \; if \; \frac{P(y_1|x_1, x_2, x_3, \ldots)}{P(y_2|x_1, x_2, x_3, \ldots)} > 1 \;\; else \; \hat{y} = y_2$$

Cheat Sheet – Bias-Variance Tradeoff

What is Bias? $bias = \mathbb{E}[f'(x)] - f(x)$

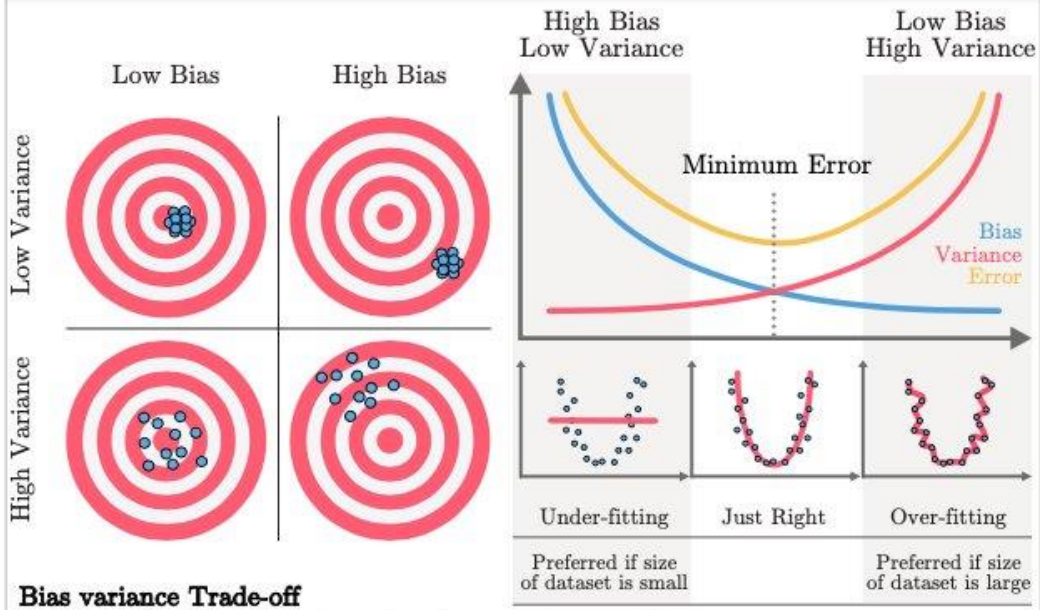
- Error between average model prediction and ground truth
- The bias of the estimated function tells us the capacity of the underlying model to predict the values

What is Variance? $variance = \mathbb{E}\left[(f'(x) - \mathbb{E}[f'(x)])^2\right]$

- Average variability in the model prediction for the given dataset
- The variance of the estimated function tells you how much the function can adjust to the change in the dataset

High Bias
- → Overly-simplified Model
- → Under-fitting
- → High error on both test and train data

High Variance
- → Overly-complex Model
- → Over-fitting
- → Low error on train data and high on test
- → Starts modelling the noise in the input



Bias variance Trade-off

- Increasing bias (not always) reduces variance and vice-versa
- Error = bias² + variance +irreducible error
- The best model is where the error is reduced.
- Compromise between bias and variance

Source: https://www.cheatsheets.aqeel-anwar.com Tutorial: Click here

Cheat Sheet – Imbalanced Data in Classification

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

Classifier that always **predicts label blue** yields prediction accuracy of 90%

Accuracy doesn't always give the correct insight about your trained model

Accuracy: %age correct prediction	Correct prediction over total predictions	One value for entire network
Precision: Exactness of model	From the detected cats, how many were actually cats	Each class/label has a value
Recall: Completeness of model	Correctly detected cats over total cats	Each class/label has a value
F1 Score: Combines Precision/Recall	Harmonic mean of Precision and Recall	Each class/label has a value

Performance metrics associated with Class 1

Predicted Labels \ Actual Labels	1	0
1	True Positive	False Positive
0	False Negative	True Negative

(Is your prediction correct?) (What did you predict)

True Negative

(Your prediction is correct) (You predicted 0)

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{F1 score} = 2x\frac{(\text{Prec} \times \text{Rec})}{(\text{Prec} + \text{Rec})}$$

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

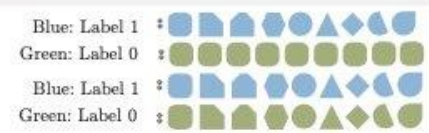
$$\text{False +ve rate} = \frac{FP}{TN + FP}$$

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

$$\text{Recall, Sensitivity} = \text{True +ve rate} = \frac{TP}{TP + FN}$$

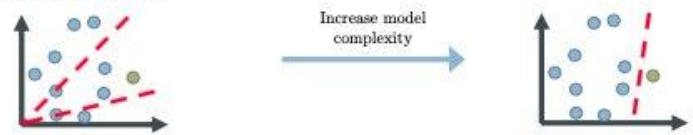
Possible solutions

1. **Data Replication:** Replicate the available data until the number of samples are comparable
2. **Synthetic Data:** Images: Rotate, dilate, crop, add noise to existing input images and create new data
3. **Modified Loss:** Modify the loss to reflect greater error when misclassifying smaller sample set



$$loss = a * loss_{green} + b * loss_{blue} \qquad a > b$$

4. **Change the algorithm:** Increase the model/algorithm complexity so that the two classes are perfectly separable (Con: Overfitting)



No straight line (y=ax) passing through origin can perfectly separate data. **Best solution:** line y=0, predict all labels blue

Straight line (y=ax+**b**) can perfectly separate data. Green class will no longer be predicted as blue

Source: https://www.cheatsheets.aqeel-anwar.com Tutorial: Click here