ANSWRSHEET 2 (Machine learning)

Answers 1 to Answers 10

Ans 1 (B) They cannot be used when the data is not completely linearly separable while allowing no errors.

Ans 2 (C) Any possible classifier which can linearly separate the data of two classes is called maximal margin classifier.

Ans 3 (C) They allow some degree of errors or misclassification.

Ans 4 (A) They take the data from lower dimensional space to some higher dimensional space in case the data is not likely to be linearly separable.

Ans 5 (C) The data product values given by the kernel functions are used to find the classifier in the higher dimensional space.

Ans 6 (C) It is a model trained using supervised learning. It can be used for classification and regression.

Ans 7 (D) All of the above

Ans 8 (C) The data is noisy and contains overlapping points.

Ans 9 (A) Misclassification would happen.

Ans 10 (B) How accurately the SVM can predict outcomes for unseen data.

ANSWERSHEET-2(PYTHON)

Answers1 to Answers 7

Ans 1 (B) struct.

Ans 2 (C) 1_no.

Ans 3 (A) in

Ans 4 (A) Left to right.

Ans 5 (C) iv-iii-ii-I

Ans 6 ()

Ans 7 (B) str

Answers 8 to Answers 10

Ans 8(A) Division and multiplication have same precedence in python

(B) Python's operators' precedence is based on PEDMAS

(D) In case of operators' having the same precedence, the one on the left side is executed first.

Ans 9 (A) abc = 1,000,000

(C) a,b,c = 1000, 2000, 3000

(D) a_b_c = 1,000,000

Ans 10

Answers 11 to Answers 13

Ans 11

S.NO.	LIST	TUPLE	SET	DICTIONARY

1.	List is a non-homogeneous data structure that stores the elements in single row and multiple rows and columns	Tuple is also a non-homogeneous data structure that stores single row and multiple rows and columns	Set data structure is also non-homogeneous data structure but stores in single row	Dictionary is also a non-homogeneous data structure which stores key value pairs
2.	List can use nested among all.	Tuple can use nested among all.	Set can use nested among all	Dictionary can use nested among all.
3.	List can be represented by [ ]	Tuple can be represented by ( )	Set can be represented by { }	Dictionary can be represented by { }
4.	Example: [1, 2, 3, 4, 5]	Example: (1, 2, 3, 4, 5)	Example: {1, 2, 3, 4, 5}	Example: {1: “a”, 2: “b”, 3: “c”, 4: “d”, 5: “e”}
5.	List can be created using list() function	Tuple can be created using tuple() function.	Set can be created using set() function	Dictionary can be created using dic() function.

6.	List allows duplicate elements	Tuple allows duplicate elements	Set will not allow duplicate elements	Dictionary doesn't allow duplicate keys.
7.	List is ordered	Tuple is ordered	Set is unordered	Dictionary is ordered (Python 3.7 and above)
8.	List is mutable i.e we can make any changes in list.	Tuple is immutable i.e we can not make any changes in tuple	Set is mutable i.e we can make any changes in set. But elements are not duplicated.	Dictionary is mutable. But Keys are not duplicated.
9.	Creating an empty list l=[]	Creating an empty Tuple t=()	Creating a set a=set()	Creating an empty dictionary d={}

Ans 12 No, strings are not mutable in python. Strings are immutable in python. Strings are immutable means they can not be change after created.

Ans 13 Python ord() function take string argument of a single Unicode character and return its integer Unicode code point value.

Example of ord () function

```
character = 'P'


# find unicode of P
unicode_char = ord(character)

print(unicode_char)

# Output: 80
```

To determine the type of a variable in Python, use the built-in type() function. In Python, everything is an object. As a result, when you use the type() function to print the type of a variable's value to the console, it returns the class type of the object.

Ans 14

In [1]:
```
# Sum of natural numbers up to num

num = 16

if num < 0:
   print("Enter a positive number")
else:
   sum = 0
   # use while loop to iterate until zero
   while(num > 0):
       sum += num
       num -= 1
   print("The sum is", sum)
```

```
The sum is 136
```

Ans 15

In [2]:
```
# Solve the quadratic equation ax**2 + bx + c = 0

# import complex math module
import cmath

a = 1
b = 5
c = 6

# calculate the discriminant
d = (b**2) - (4*a*c)

# find two solutions
sol1 = (-b-cmath.sqrt(d))/(2*a)
sol2 = (-b+cmath.sqrt(d))/(2*a)

print('The solution are {0} and {1}'.format(sol1,sol2))
```

```
The solution are (-3+0j) and (-2+0j)
```

In [ ]:
```

```

ANSWERSHEET-2 (STATISTICS)

Answers 1 to Answers 12

Ans 1 (C) Type I; Type II

Ans 2 (B) We have made a correct decision

Ans 3 (B) critical value

Ans 4 (B) A Type I error was made.

Ans 5 (C) $x = 17, s = 7$

Ans 6 (A) fail to reject H0

Ans 7 (C) At $\alpha = 0.05$, reject the null hypothesis.

Ans 8 (B)0.041

Ans 9 (C) 0.958

Ans 10 (C). Left tail

Ans 11 (A) . Less than the significance level

Ans 12 (B) 0.375

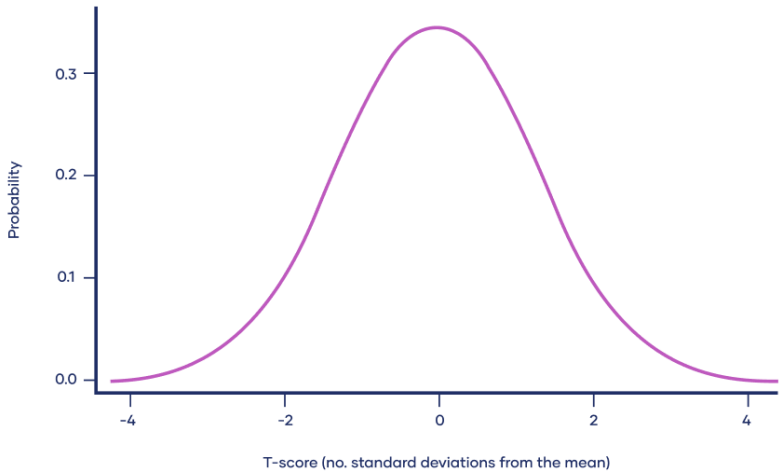
ANSWERS 13 TO ANSWERS 15

ANS 13

The *t*-distribution is a type of normal distribution that is used for smaller sample sizes. Normally-distributed data form a bell shape when plotted on a graph, with more observations near the mean and fewer observations in the tails.

The *t*-distribution is used when data are *approximately* normally distributed, which means the data follow a bell shape but the population variance is unknown. The variance in a *t*-distribution is estimated based on the degree of freedom of the data set (total number of observations minus 1).

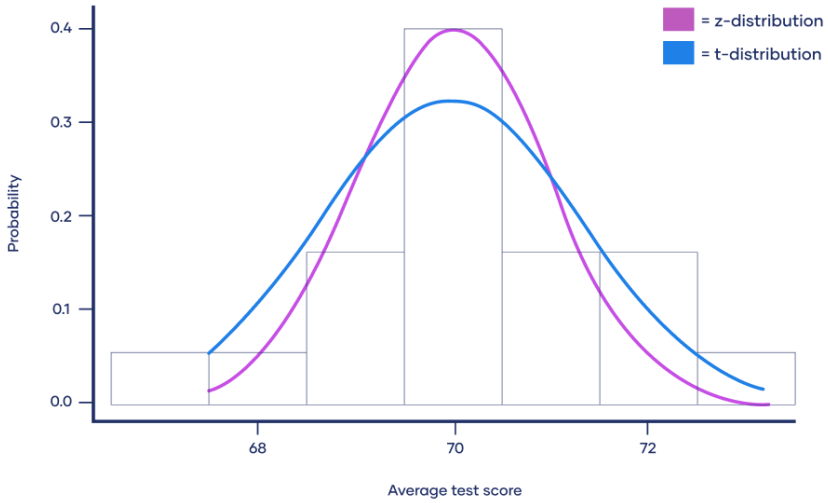
.It is a type of normal distribution used for smaller sample sizes, where the variance in the data is unknown.



It is a more conservative form of the standard normal distribution, also known as the *z*-distribution. This means that it gives a lower probability to the center and a higher probability to the tails than the standard normal distribution

EXAMPLES OF T-distribution and Z- distribution

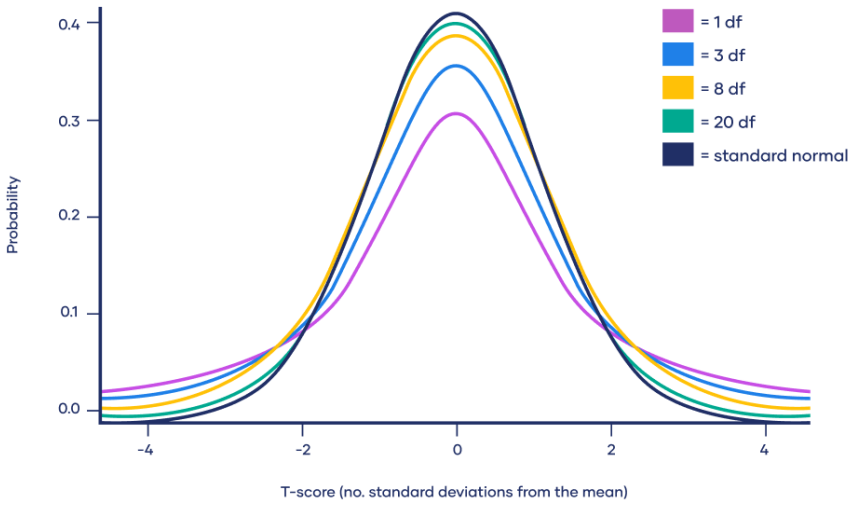
If we measure the average test score from a sample of only 20 students, we should use the *t*-distribution to estimate the confidence interval around the mean. If we use the *z*-distribution, our confidence interval will be artificially precise.



As the degree of freedom(total number of observations minus 1) increases, the *t*-distribution will get closer and closer to matching the standard normal distribution, a.k.a. the *z*-distribution, until they are almost identical.

Above 30 degrees of freedom, the *t*-distribution roughly matches the *z*-distribution. Therefore, the *z*-distribution can be used in place of the *t*-distribution with large sample sizes.

The *z*-distribution is preferable over the *t*-distribution when it comes to making statistical estimates because it has a known variance. It can make more precise estimates than the *t*-distribution, whose variance is approximated using the degrees of freedom of the data.



Ans 14

Yes, the T- distribution is normal distribution that is used for smaller sample sizes. Normally-distributed data form a bell shape when plotted on a graph, with more observations near the mean and fewer observations in the tails.

Ans 15

The t-distribution describes the standardized distances of sample means to the population mean when the population standard deviation is not known, and the observations come from a normally distributed population.