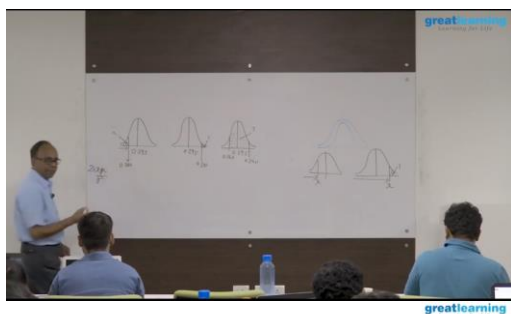


DAILY ASSESSMENT FORMAT

| | | | |
|-----------------------|--------------------------------------|------------------------|------------|
| Date: | 17/06/2020 | Name: | Lavanya B |
| Course: | Statistical learning | USN: | 4a117ec043 |
| Topic: | Bayes Theorem Normal distribution | Semester & Section: | 6th A |
| Github Repository: | Lavanya-B | | |

FORENOON SESSION DETAILS

Image of session



Normal Probability Density Function

In the usual notation, the probability density function of the normal distribution is given below:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left[\frac{(x-\mu)^2}{2\sigma^2}\right]}$$

x is a continuous normal random variable with the property $-\infty < x < \infty$ meaning x can take all real numbers in the interval $-\infty < x < \infty$.

Report

Bayes Theorem

Bayes' theorem thus gives the probability of an event based on new information that is, or may be related, to that event. The formula can also be used to see how the probability of an event occurring is affected by hypothetical new information, supposing the new information will turn out to be true.

KEY TAKEAWAYS

- Bayes' theorem allows you to update predicted probabilities of an event by incorporating new information.
- Bayes' theorem was named after 18th-century mathematician Thomas Bayes.
- It is often employed in finance in updating risk evaluation.

Normal distribution

Normal distribution, also known as the Gaussian distribution, is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean. In graph form, normal distribution will appear as a bell curve.

In probability theory, a normal distribution is a type of continuous probability distribution for a real-valued random variable. The general form of its probability density function is $\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. The parameter μ is the mean or expectation of the distribution; and σ is its standard deviation.

Parameters of Normal Distribution

1. **Mean-** The mean is the central tendency of the distribution. It defines the location of the peak for normal distributions. Most values cluster around the mean. On a graph, changing the mean shifts the entire curve left or right on the X-axis.
2. **Standard deviation-** The standard deviation is a measure of variability. It defines the width of the normal distribution. The standard deviation determines how far away from the mean the values tend to fall. It represents the typical distance between the observations and the average.

On a graph, changing the standard deviation either tightens or spreads out the width of the distribution along the X-axis. Larger standard deviations produce distributions that are more spread out.

Course completion certificate

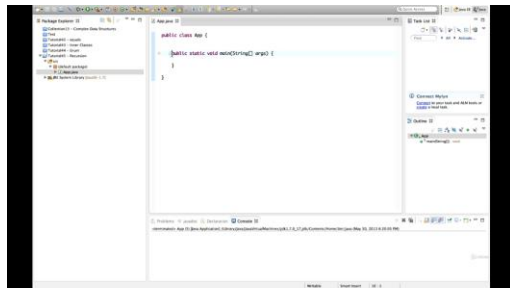


Date: 17/06/2020
Course: Udemy
Topic: JAVA

Name: Lavanya B
USN: 4a17ec043
Semester 6th A
& Section:

AFTERNOON SESSION DETAILS

Image of session



Lectures

More

- 40 Video - 08:49 mins - Resources (1)
- 41 ☒ Abstract Classes
Video - 12:58 mins - Resources (1)
- 42 ☒ Reading Files With File Reader
Video - 17:26 mins - Resources (1)
- 43 ☒ Try-With-Resources
Video - 11:12 mins - Resources (1)
- 44 ☒ Creating and Writing Text Files
Video - 06:20 mins - Resources (1)
- 45 ☒ The Equals Method
Video - 17:21 mins - Resources (1)
- 46 ☒ Inner Classes
Video - 16:33 mins - Resources (1)
- 47 ☒ Enum Types: Basic and Advanced Usage
Video - 19:20 mins - Resources (1)
- 48 ☒ Recursion: A Useful Trick Up Your Sleeve
Video - 17:26 mins - Resources (1)

Report

Sets

A Set is a collection that cannot contain duplicate elements. It models the mathematical set abstraction.

One of the implementations of the Set is the HashSet class.

The HashSet class does not automatically retain the order of the elements as they're added. To order the elements, use a LinkedHashSet, which maintains a linked list of the set's elements in the order in which they were inserted.

Eg:

```
import java.util.HashSet;

public class MyClass {
    public static void main(String[ ] args) {
        HashSet<String> set = new HashSet<String>();
        set.add("A");
        set.add("B");
        set.add("C");
        System.out.println(set);
    }
}
```

Sorting list

For the manipulation of data in different collection types, the Java API provides a Collections class, which is included in the java.util package.

One of the most popular Collections class methods is sort(), which sorts the elements of your collection type. The methods in the Collections class are static, so you don't need a Collections object to call them.

Eg:

```
import java.util.ArrayList;
import java.util.Collections;

public class MyClass {
    public static void main(String[ ] args) {
        ArrayList<Integer> nums = new ArrayList<Integer>();
        nums.add(3);
        nums.add(36);
        nums.add(73);
        nums.add(40);
        nums.add(1);
    }
}
```

```
    Collections.sort(nums);
    System.out.println(nums);
}
}
```

Iterators

An Iterator is an object that enables to cycle through a collection, obtain or remove elements. Before you can access a collection through an iterator, you must obtain one. Each of the collection classes provides an `iterator()` method that returns an iterator to the start of the collection. By using this iterator object, you can access each element in the collection, one element at a time.

The Iterator class provides the following methods:

`hasNext()`: Returns true if there is at least one more element; otherwise, it returns false.

`next()`: Returns the next object and advances the iterator.

`remove()`: Removes the last object that was returned by `next` from the collection.

Eg:

```
import java.util.Iterator;
```

```
import java.util.LinkedList;
```

```
public class MyClass {
    public static void main(String[ ] args) {
        LinkedList<String> animals = new LinkedList<String>();
        animals.add("fox");
        animals.add("cat");
        animals.add("dog");
        animals.add("rabbit");

        Iterator<String> it = animals.iterator();
        while(it.hasNext()) {
            String value = it.next();
            System.out.println(value);
        }
    }
}
```

Files

The `java.io` package includes a `File` class that allows you to work with files. create a `File` object and specify the path of the file in the constructor.

Files are useful for storing and retrieving data, and there are a number of ways to read from a file. One of the simplest ways is to use the `Scanner` class from the `java.util` package. The constructor of the `Scanner` class can take a `File` object as input.

Eg:

```
import java.io.File;
import java.util.Scanner;
import java.util.Formatter;

public class MyClass {
    public static void main(String[ ] args) {
        try {
            Formatter f = new Formatter("test.txt");
            f.format("%s %s %s", "1", "John", "Smith \r\n");
            f.format("%s %s %s", "2", "Amy", "Brown");
            f.close();

            File x = new File("test.txt");
            Scanner sc = new Scanner(x);
            while(sc.hasNext()) {
                System.out.println(sc.next());
            }
            sc.close();
        } catch (Exception e) {
            System.out.println("Error");
        }
    }
}
```

Webinar

Topic: Break the Chain Socially & Connect the Chain by Building Blocks & Career



Industry Specific Examples



| Industry | Examples |
|------------------------------|---|
| BFSI | American Express, Bank of America, ICICI Bank , Axis, SBI , Kotak Mahindra Bank |
| Manufacturing & Supply Chain | Walmart, Alibaba, Maersk |
| Healthcare | FarmaTrust, Apollo Hospitals, Strides Pharmacy |
| Automobile | Mercedes Benz, Daimler, Mahindra, Hero Group |
| Public Sector | USPS, US Dept. Homeland Security, NITI Aayog , Coffee Board of India, Government of Telangana |
| Energy & Utilities | Reliance Power, Shell, Essentia.one |

skill@emurgo.in

Rapidly Growing Roles



Key Roles for Techies

- Blockchain DApp Developers
- Blockchain Protocol Engineers
- Blockchain Network Administrators
- Blockchain Architects / Project Managers
- Blockchain Quality Engineers

Key Non Developer Roles

- Blockchain Product Manager
- Blockchain Business/Financial Analyst
- Blockchain Business Consultant
- Blockchain Evangelist



skill@emurgo.in