**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **16/06/2020** | **Name:** | **Lavanya B** |
| **Course:** | **Statical learning** | **USN:** | **4al17ec043** |
| **Topic:** | **Case study on statistics and probability Theory**  **Introduction and rules for probability** | **Semester & Section:** | **6th A** |
| **Github Repository:** | **Lavanya-B** |  |  |

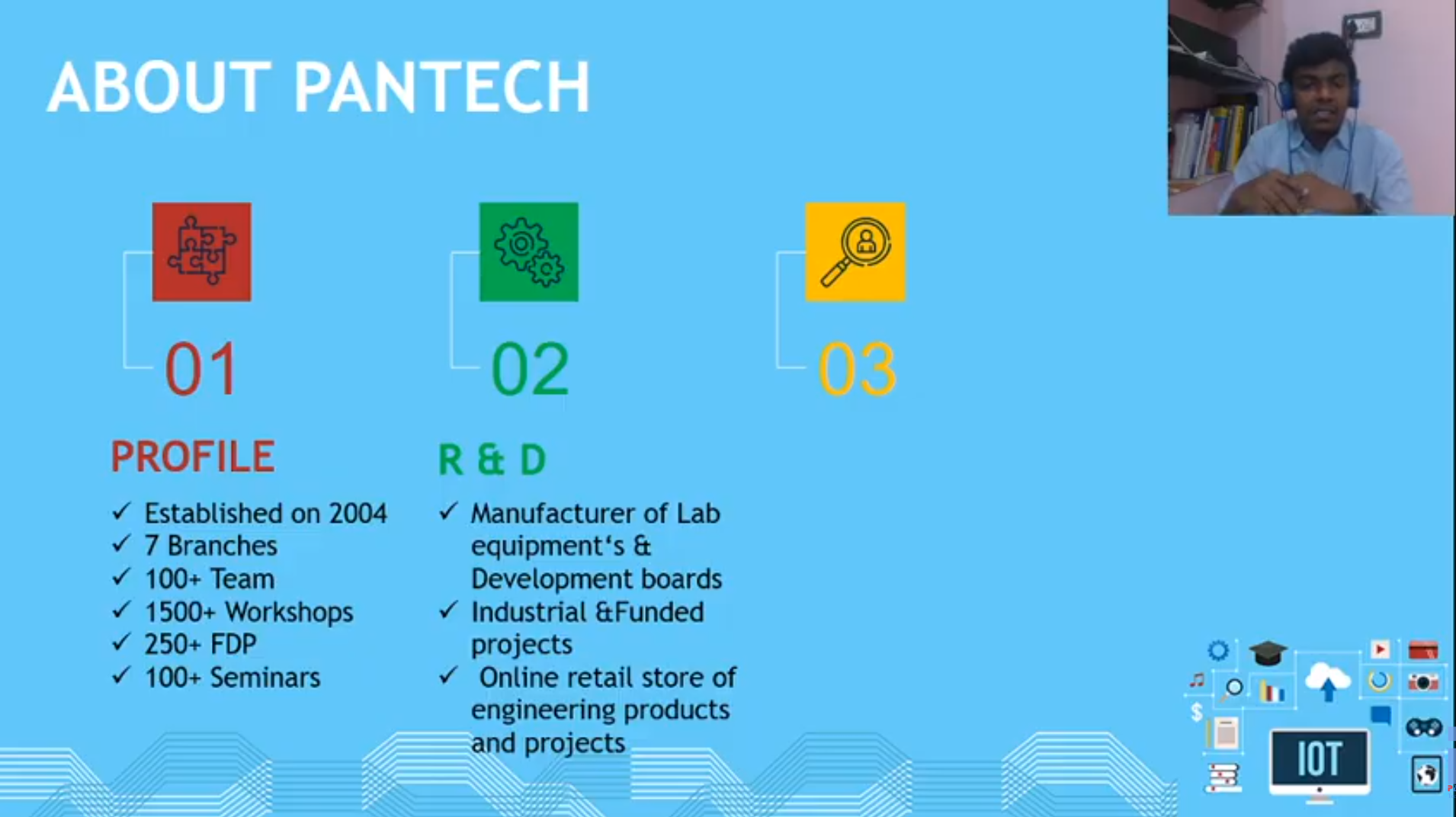
|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report**  **Statical learning**  **Statistical learning theory is a framework for machine learning drawing from the fields of statistics and functional analysis. Statistical learning theory deals with the problem of finding a predictive function based on data.**  **This is an introductory-level course in supervised learning, with a focus on regression and classification methods. The syllabus includes: linear and polynomial regression, logistic regression and linear discriminant analysis; cross-validation and the bootstrap, model selection and regularization methods, nonlinear models, splines and generalized additive models; tree-based methods, random forests and boosting; support-vector machines. Some unsupervised learning methods are discussed: principal components and clustering.**  **Statistics and Probability Theory**  **Probability theory is the branch of mathematics concerned with probability. Although there are several different probability interpretations, probability theory treats the concept in a rigorous mathematical manner by expressing it through a set of axioms.**   * **Probability of combination of events** * **Probabilities and random variables** * **Distribution of random variables**   **In making the transition to finite sample sizes we also make the transition from the theoretical realm**  **of probability theory to the more practical world of statistical analysis. Thus we should spend some time**  **understanding the basic tenets of statistics before we use the results.**   * **Moments of binomial distribution** * **Multiple variables, variance and covariance** * **Maximum likelihood**   **Rules for Probability calculation**   1. **Probability Rule One (For any event A, 0 ≤ P(A) ≤ 1)** 2. **Probability Rule Two (The sum of the probabilities of all possible outcomes is 1)** 3. **Probability Rule Three (The Complement Rule)** 4. **Probabilities Involving Multiple Events.** 5. **Probability Rule Four (Addition Rule for Disjoint Events)** 6. **Finding P(A and B) using Logic.** |

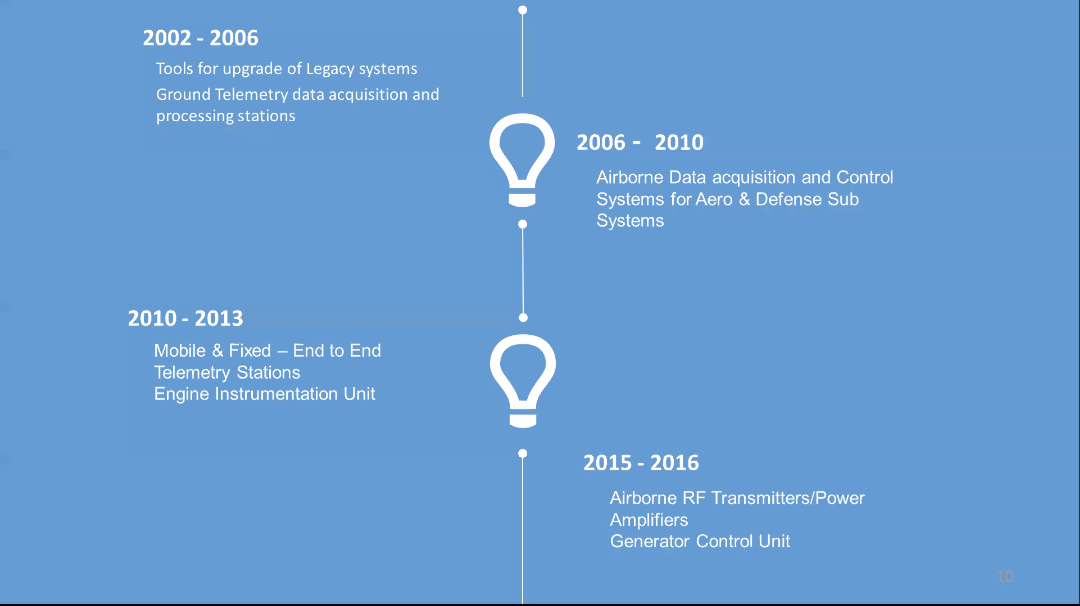
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **16/06/2020** | **Name:** | **Lavanya B** | |
| **Course:** | **JAVA** | **USN:** | **4al17ec043** | |
| **Topic:** | **The JAVA collection frame work** | **Semester & Section:** | **6th A** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report**  **Array lists**  **The Java API provides special classes to store and manipulate groups of objects.**  **One such class is the ArrayList. Standard Java arrays are of a fixed length, which means that after they are created, they cannot expand or shrink.**  **On the other hand, ArrayLists are created with an initial size, but when this size is exceeded, the collection is automatically enlarged.**  **When objects are removed, the ArrayList may shrink in size. Note that the ArrayList class is in the java.util package, so it's necessary to import it before using it.**  **The ArrayList class provides a number of useful methods for manipulating its objects.**  **The add() method adds new objects to the ArrayList. Conversely, the remove() method removes objects from the ArrayList.**  **Eg:**  **import java.util.ArrayList;**  **public class MyClass {**  **public static void main(String[ ] args) {**  **ArrayList<String> colors = new ArrayList<String>();**  **colors.add("Red");**  **colors.add("Blue");**  **colors.add("Green");**  **colors.add("Orange");**  **colors.remove("Green");**    **System.out.println(colors);**  **}**  **}**  **Linked list**  **The LinkedList is very similar in syntax to the ArrayList.**  **You can easily change an ArrayList to a LinkedList by changing the object type. The LinkedList is better for manipulating data, such as making numerous inserts and deletes.**  **In addition to storing the object, the LinkedList stores the memory address (or link) of the element that follows it. It's called a LinkedList because each element contains a link to the neighboring element.**  **Eg:**  **import java.util.LinkedList;**  **public class MyClass {**  **public static void main(String[ ] args) {**  **LinkedList<String> c = new LinkedList<String>();**  **c.add("Red");**  **c.add("Blue");**  **c.add("Green");**  **c.add("Orange");**  **c.remove("Green");**    **for(String s: c) {**  **System.out.println(s);**  **}**  **}**  **}**  **Hashmap**  **HashMap is used for storing data collections as key and value pairs. One object is used as a key (index) to another object (the value).**  **The put, remove, and get methods are used to add, delete, and access values in the HashMap.**  **A HashMap cannot contain duplicate keys. Adding a new item with a key that already exists overwrites the old element.**  **The HashMap class provides containsKey and containsValue methods that determine the presence of a specified key or value.**  **If you try to get a value that is not present in your map, it returns the value of null.**  **Eg:**  **import java.util.HashMap;**  **public class MyClass {**  **public static void main(String[ ] args) {**  **HashMap<String, Integer> points = new HashMap<String, Integer>();**  **points.put("Abc", 154);**  **points.put("Def", 42);**  **points.put("Ghi", 733);**  **System.out.println(points.get("Def"));**  **}**  **}** | | | |

**Webinar**

**Future Ahead for Electronics Engineers**

**Speaker: Mr. Radhakrishnan**

****

****

****