**Learning Artificial intelligence and Machine learning**

**COURSE IN INFOSYS SPRINGBOARD**

**Infosys Springboard**

****

include Infosys Springboard certificates on your CV under a relevant "Certifications" or "Professional Development" section to showcase additional skills and training.Infosys Springboard is unlocking the doors to world-class digital education and democratizing learning for the leaders of tomorrow.

Infosys Springboard brings digital and life skills for students from class 6 to lifelong learners in India. Delivered free through Infosys CSR commitment to empowering 10 million-plus people with digital skills by 2025, it provides a holistic set of courses,

developed in collaboration with world-leading digital educators like Coursera and Harvard Business Publishing. It is fully aligned with India’s National Education Policy 2020.

The learning program is particularly well-suited to grow vocational skills in addition to soft skills. Masterclasses, programming challenges, practice areas, and playgrounds for experimentation make the training immersive for all. The assessment formats that follow are new-age ready with certification for learners who pass the virtual proctored examination. Nearly 400,000 learners and 300+ education institutions, NGOs, and support groups are already onboard Infosys Springboard.

include Infosys Springboard certificates on your CV under a relevant "Certifications" or "Professional Development" section to showcase additional skills and training.Highlight how these certificates contribute to your qualifications for the desired role.

include Infosys Springboard certificates on your CV under a relevant "Certifications" or "Professional Development" section to showcase additional skills and training. Highlight how these certificates contribute to your qualifications for the desired role.

Experience AI-powered, immersive labs, secure assessments, and social learning. Access 8,000+ FREE courses to unlock your potential for industry needs.

This is a free online platform powered by Infosys Wingspan. It offers guided training in technology and curated content to address reskilling of unemployed in tech, non tech and support roles.

**Course overview**

|  |  |  |  |
| --- | --- | --- | --- |
| Program | Computer science and Engineering | Course Duration | 6h 0m |
| Course Name | Artificial intelligence and Machine learning | Course Level | Beginner |
| Course Provider | Infsoys Springboard | **Auth**or | Ben Fhala |

**Course Outcome**



Course outcomes for a typical Artificial Intelligence (AI) and Machine Learning (ML) course can vary depending on the level (undergraduate or graduate) and the specific focus of the course. However, here are some common learning outcomes you might expect from such a course:

**1.Understanding of Fundamentals**:

* Explain the basic principles and concepts of artificial intelligence and machine learning.
* Define key terminologies such as supervised learning, unsupervised learning, reinforcement learning, neural networks, etc.

**2.Mathematical Foundation**

* Apply linear algebra, calculus, probability theory, and statistics to understand and analyze machine learning algorithms.

**3.Machine Learning Algorithms**:

* Describe and implement common machine learning algorithms such as linear regression, logistic regression, decision trees, support vector machines, k-nearest neighbors, clustering algorithms, etc.
* Understand the strengths, weaknesses, and appropriate use cases for different algorithms.

**4.Data Preprocessing and Feature Engineering**:

* Clean and preprocess datasets for machine learning tasks.
* Perform feature selection and extraction to improve model performance.

**5.Model Evaluation and Validation**:

* Evaluate machine learning models using appropriate metrics (accuracy, precision, recall, F1-score, ROC curves, etc.).
* Apply techniques such as cross-validation and regularization to improve model generalization.

**6.Deep Learning**:

* Understand the basics of neural networks and deep learning.
* Implement and train deep learning models using frameworks like TensorFlow or PyTorch.

**7.Applications of AI and ML**:

* Identify real-world applications of AI and ML across various domains such as healthcare, finance, natural language processing, computer vision, etc.
* Discuss ethical considerations and societal impacts of AI and ML technologies.

**8.Hands-on Programming and Projects**:

* Implement machine learning algorithms and models using programming languages like Python.
* Work on projects to solve real-world problems using AI and ML techniques.

**9.Critical Thinking and Problem Solving**:

* Critically evaluate different approaches to solving a given problem using AI and ML techniques.
* Develop strategies for improving model performance and addressing challenges encountered in practical applications.

**10.Communication Skills**:

* Effectively communicate the results and findings of machine learning experiments and projects.
* Present technical concepts to both technical and non-technical audiences.

These outcomes collectively aim to equip students with both theoretical knowledge and practical skills necessary to apply AI and ML techniques effectively in various domains and industries.

**Table of the Content**

|  |  |  |
| --- | --- | --- |
| **Slno.** | **Content Name** | **Duration** |
| **1** | Prelude   * Introduction to AI * Topic coverage * Learning outcome | 2m 28s |
| **2** | Why AI   * Why AI? * Case Studies - part 1 * Case Studies - part 2 * AI Business Opportunities * Business value of AI * Case Studies - Quiz | 37m 28s |
| **3** | Why is AI   * History of AI * Type of AI * Aspects of AI -part 1 * Aspects of AI -part * Technical view of AI * Machine Learning * How does a machine make prediction * Supervised Learning * Types of supervised learning * Classification * Support vector machine(SVM) * Decision tree * K – nearest neighbors(KNN) * Logistic regression * Regression * Deep learning – part 1 * Deep learning – part 2 * Deep learning – part 3 * Type of artificial neural Network(ANN) * Overfitting and undefitting | 57m 38s |
| **4** | AI in practice   * Machine learning process – part 1 * Machine learning process – part 2 * Machine learning use case * Use case * Use case – Quiz * AI Architecture * AI Architecture – Quiz * AI led business process transformations * AI led business process transformation * AI led business process transformation - Quiz | 43m 5s |
| **5** | Mango DB and Beyond   * Installing mango DB * Creating a mango database * Inserting collection and documents * Adding is great,but how do you find things * Finding complicated things with operators * Deleting documents,collection * Assessment | 1h 38m |
| **6** | Ruby fullstack development   * Introduction to ruby * Install and configure ruby * Create and run a simple ruby app * Interactive ruby * Ruby identifies * Using string,numerical and Boolean values * Applying comments and structure to ruby code * Work with operators on variables * Control input and output * Use flow control * Use logical operator * Work with blocks,procs and lambdas * Identify variables and namespace scope | 4h 5s |
| **7** | **Threads and security**   * Creating and using a thread * Working with thread variables and thread keys * Recognizing thread priorities and using mutexes * Identifying race conditions and deadlocks * Recognizing ruby lang security vulnerabilities * Handling nil objects * Creating immutable methods for safety * Using symmetric ciphers * Using hash function * Creating secure random numbers * Generating secure passwords | 3h 58m |

**Node.js**

**Node.js** is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform), [open-source](https://en.wikipedia.org/wiki/Open-source_software) server environment that can run on [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://en.wikipedia.org/wiki/Linux), [Unix](https://en.wikipedia.org/wiki/Unix), [macOS](https://en.wikipedia.org/wiki/MacOS), and more. Node.js is a [back-end](https://en.wikipedia.org/wiki/Frontend_and_Backend) [JavaScript](https://en.wikipedia.org/wiki/JavaScript) [runtimeenvironment](https://en.wikipedia.org/wiki/Runtime_system), runs on the [V8](https://en.wikipedia.org/wiki/V8_(JavaScript_engine)) [JavaScriptengine](https://en.wikipedia.org/wiki/JavaScript_engine), and executes JavaScript code outside a [webbrowser](https://en.wikipedia.org/wiki/Web_browser).

Node.js lets developers use JavaScript to write command line tools and for [server-sidescripting](https://en.wikipedia.org/wiki/Server-side_scripting). The ability to run JavaScript code on the server is often used to generate [dynamicwebpage](https://en.wikipedia.org/wiki/Dynamic_web_page) content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm,unifying [web-application](https://en.wikipedia.org/wiki/Web_application) development around a single programming language, as opposed to using different languages for the server- versus client-side programming.

Node.js has an [event-drivenarchitecture](https://en.wikipedia.org/wiki/Event-driven_architecture) capable of [asynchronous I/O](https://en.wikipedia.org/wiki/Asynchronous_I/O). These design choices aim to optimize [throughput](https://en.wikipedia.org/wiki/Throughput) and [scalability](https://en.wikipedia.org/wiki/Scalability) in web applications with many input/output operations, as well as for [real-timeWeb](https://en.wikipedia.org/wiki/Real-time_Web) applications (e.g., [real-timecommunication](https://en.wikipedia.org/wiki/Real-time_communication) programs and [browsergames](https://en.wikipedia.org/wiki/Browser_game)).

The Node.js [distributeddevelopment](https://en.wikipedia.org/wiki/Distributed_development) project was previously governed by the Node.js Foundation, and has now merged with the [JSFoundation](https://en.wikipedia.org/wiki/JS_Foundation) to form the [OpenJSFoundation](https://en.wikipedia.org/wiki/OpenJS_Foundation). OpenJS Foundation is facilitated by the [LinuxFoundation](https://en.wikipedia.org/wiki/Linux_Foundation)'s Collaborative Projects program.

**Express.js**

Express.js is a small framework that works on top of Node.js web server functionality to simplify its APIs and add helpful new features. It makes it easier to organize your application’s functionality with middleware and routing. It adds helpful utilities to Node.js HTTP objects and facilitates the rendering of dynamic HTTP objects.

**Mango DB**

MongoDB is an open source [NoSQL](https://www.techtarget.com/searchdatamanagement/definition/NoSQL-Not-Only-SQL) database management program. NoSQL (Not only SQL) is used as an alternative to traditional relational databases. NoSQL databases are quite useful for working with large sets of distributed data. MongoDB is a tool that can manage document-oriented information, store or retrieve information.

MongoDB is used for high-volume data storage, helping organizations store large amounts of data while still performing rapidly. Organizations also use MongoDB for its ad-hoc queries, indexing, [loadbalancing](https://www.techtarget.com/searchnetworking/definition/load-balancing), aggregation, server-side JavaScript execution and other features.

Structured Query Language ([SQL](https://www.techtarget.com/searchdatamanagement/definition/SQL)) is a standardized programming language that is used to manage relational databases. SQL normalizes data as schemas and tables, and every table has a fixed structure.

Instead of using tables and rows as in [relationaldatabases](https://www.techtarget.com/searchdatamanagement/definition/relational-database), as a NoSQL database, the MongoDB architecture is made up of collections and documents. Documents are made up of [key-valuepairs](https://www.techtarget.com/searchenterprisedesktop/definition/key-value-pair) -- MongoDB's basic unit of data. Collections, the equivalent of SQLtables, contain document sets. MongoDB offers support for many programming languages, such as C, C++, C#, Go, Java, Python, Ruby and Swift.

**Ruby**

Ruby is a very powerful language for web and software development. During this course, you can explore how to code in Ruby and the basic building blocks of a Ruby web application. Learn to install and configure Ruby, create and run a simple Ruby app, and operate Interactive Ruby (IRB). Recognize key Ruby identifiers, such as variables and constants. Use string, numeric, and Boolean values. Apply comments and structure to your Ruby programs, along with arithmetic and string operators, to manipulate variables. Use Ruby methods to get user input and format output to the console. Add flow control, logical operators, blocks, procs, and lambdas. Conclude by identifying how variable scope and namespaces work in Ruby.

Full stack Ruby on rails refers to a developer who use implements Ruby on Rails language in backend web development. Generally , full stack is referred to developes who handles front end development, backend development, databases , and handles servers.

**Thread in Ruby**

A new threads are created with Thread.new. You can also use the synonyms Thread.start and Thread*.*fork.

There is no need to start a thread after creating it, it begins running automatically when CPU resources become available.

The Thread class defines a number of methods to query and manipulate the thread while it is running. A thread runs the code in the block associated with the call to Thread.new and then it stops running.

The value of the last expression in that block is the value of the thread, and can be obtained by calling the value method of the Thread object. If the thread has run to completion, then the value returns the thread's value right away. Otherwise, the value method blocks and does not return until the thread has completed.

The class method Thread.current returns the Thread object that represents the current thread. This allows threads to manipulate themselves. The class method Thread.main returns the Thread object that represents the main thread. This is the initial thread of execution that began when the Ruby program was started.

You can wait for a particular thread to finish by calling that thread's Thread.join method. The calling thread will block until the given thread is finished.

**Course Summary**

In this 12 hour 15 minutes course we learnt about the learning full stack . A full-stack developer is an IT professional who develops a software application's client and server side. They apply a wide range of coding skills and help other programmers solve problems. Full-stack developers also test and debug software and optimize the code.



****

****

****

****

**Feedback**

****

Overall springboard was a great experience and it was very supportive. All topics were in deep it was very helpful to us we have learnt many topics thank you for providing this wonderful course.

