Machine Learning

Duration: 72 Hours, 3 Days per week, 2 Hours per day

Module	Major Categories	Duration
Module 01	Programming with Python and Related Libraries of Machine Learning	12 Hours
Module 02	Traditional Machine Learning Algorithms and Projects	24 Hours
Module 03	Concepts and Uses of Deep Learning and Real Life Projects	36 Hours

Module 01: Programming with Python and Related Libraries of Machine Learning

Section	Topics		Duration
1.1	**Introduction to Python and Programming Fundamentals**		02 Hours
	1.1.1	Basic syntax of Python and data types	
	1.1.2	Variables, Basic arithmetic and logical operations	
	1.1.3	Hands-on exercises for basic data type and variable operations	
1.2	**Conditional Statement in Python**		02 Hours
	1.2.1	Syntax of if statements	
	1.2.2	Syntax of if-else statements	
	1.2.3	Syntax of nested if statements	
	1.2.4	Syntax of elif statements	
1.3	**Introduction to Iteration and Advanced Iteration Techniques**		02 Hours
	1.3.1	Basic syntax and examples of for and while loop in Python	
	1.3.2	Nested Loops and their applications, Loop control statements (i.e., break and continue)	
	1.3.3	Range function and its usage in loops	

1.4	**Introduction to Python Functions**		02 Hours
	1.4.1	Overview of functions in Python, Creating and calling a function	
	1.4.2	Function arguments and parameters, Default parameters	
	1.4.3	Importing modules and libraries, Using functions from libraries	
	1.4.4	Understanding the Python standard library and popular third-party libraries	
1.5	**Data Structures**		02 Hours
	1.5.1	Understanding the difference between lists and tuples, Creating, indexing and slicing lists and tuples	
	1.5.2	Introduction to dictionaries and their use, Common dictionary methods like keys(), values(), items() and so on.	
	1.5.3	Introduction to sets, Set operations: union(), intersection(), difference() and symmetric_difference()	
1.6	**Concepts of Machine Learning Related Libraries of Python**		02 Hours
	1.6.1	Introduction to Numpy arrays and their use.	
	1.6.2	Introduction to Pandas DataFrames and their use	
	1.6.3	Introduction to Matplotlib for data visualization	
	1.6.4	Overview to scikit-learn for machine learning algorithms	

Module 02: Traditional Machine Learning Algorithms and Projects

Section	Topics			Duration
2.1	**Background of Machine Learning**			02 Hours
	2.1.1	Overview of Machine Learning		
	2.1.2	Categories of Machine Learning Algorithm		
	2.1.3	Parametric and Non-Parametric Machine Learning Algorithm		
	2.1.4	Supervised, Unsupervised and Semi-supervised Learning		
		2.1.4.1	Difference among Regression, Classification, Clustering and Dimensionality Reduction Technique in Machine Learning	
	2.1.5	The Bias-Variance Trade-Off		
	2.1.6	Overfitting and Underfitting		
	2.1.7	How to Detect Overfitting and Underfitting?		
	2.1.8	Reasons for Overfitting anf Underfitting		
	2.1.9	How to Prevent Overfitting and Underfitting?		
2.2	**Popular Machine Learning Algorithms**			02 Hours
	2.2.1	Linear Machine Learning Algorithms		
		2.2.1.1	Linear Regression	
		2.2.1.2	Logistic Regression	
		2.2.1.3	Linear Discriminant Analysis	
	2.2.2	Non-Linear Machine Learning Algorithms		
		2.2.2.1	Classification and Regression Trees	
		2.2.2.2	Naive Bayes	
		2.2.2.3	K Nearest Neighbour	
		2.2.2.4	K-Means	
		2.2.2.5	Linear Vector Quantization	
		2.2.2.6	Support Vector Machine	
	2.2.3	Ensemble Algorithms		
		2.2.3.1	Bagging and Random Forest	
		2.2.3.2	Boosting and AdaBoost	
2.3	**Implementation of Machine Learning Algorithms in Python**			

	2.3.1	How To Load Machine Learning Data			02 Hours
		2.3.1.1	Load CSV File with NumPy		
		2.3.1.2	Load CSV File with Pandas		
	2.3.2	Understand Your Data with Descriptive Statistics			
		2.3.2.1	Peek the Data		
		2.3.2.2	Dimensions of the Data		
		2.3.2.3	Data Type of Each Attribute		
		2.3.2.4	Descriptive Statistics		
		2.3.2.5	Number of Class Distribution		
		2.3.2.6	Correlations between Attributes		
		2.3.2.7	Skew of Univariate Distributions		
	2.3.3	Understand Your Data with Visualization			02 Hours
		2.3.3.1	Univariate Plots		
			3.3.1.1	Histogram	
			3.3.1.2	Density Plots	
			3.3.1.3	Box and Whisker Plots	
		2.3.3.2	Multivariate Plots		
			3.3.2.1	Correlation Matrix Plot	
			3.3.2.2	Scatter Plot Matrix	
	2.3.4	Data Pre-processing			02 Hours
		2.3.4.1	Needs for Data Pre-processing		
		2.3.4.2	Rescale Data		
		2.3.4.3	Standardize Data		
		2.3.4.4	Normalize Data		
		2.3.4.5	Binarize Data (Make Binary)		
	2.3.5	Feature Selection for Machine Learning			02 Hours
		2.3.5.1	Needs for Feature Selection		
		2.3.5.2	Univariate Selection		
		2.3.5.3	Recursive Feature Elimination		
		2.3.5.4	Principal Component Analysis		
		2.3.5.5	Feature Importance using Bagged decision trees		
	2.3.6	Train and Test Dataset Split			
		2.3.6.1	Split into Train and Test Sets		
		2.3.6.2	K-fold Cross Validation		

		2.3.6.3	Leave One Out Cross Validation			
		2.3.6.4	Repeated Random Test-Train Splits			
	2.3.7	Performance Matrix				02 Hours
		2.3.7.1	Classification Metrics			
			2.3.7.1.1	Classification Accuracy		
			2.3.7.1.2	Logarithmic Loss		
			2.3.7.1.3	Area Under ROC Curve		
			2.3.7.1.4	Confusion Matrix		
			2.3.7.1.5	Classification Report		
		2.3.7.2	Regression Metrics			
			2.3.7.2.1	Mean Absolute Error		
			2.3.7.2.2	Mean Squared Error		
			2.3.7.2.3	R Squared Metric		
	2.3.8	Spot-Check Classification Algorithms				02 Hours
		2.3.8.1	Linear Classification Algorithms			
			2.3.8.1.1	Logistic Regression		
			2.3.8.1.2	Linear Discriminant Analysis		
		2.3.8.2	Non-linear Classification Algorithms			
			2.3.8.2.1	k-Nearest Neighbors		
			2.3.8.2.2	Naive Bayes		
			2.3.8.2.3	Classification and Regression Trees		
			2.3.8.2.4	Support Vector Machines		
	2.3.9	Spot-Check Regression Algorithms				02 Hours
		2.3.9.1	Linear Regression Algorithms			
			2.3.9.1.1	Linear Regression		
			2.3.9.1.2	Ridge Regression		
			2.3.9.1.3	LASSO Linear Regression		
			2.3.9.1.4	Elastic Net Regression		
		2.3.9.2	Non-linear Regression Algorithms			
			2.3.9.2.1	k-Nearest Neighbors		
			2.3.9.2.2	Classification and Regression Trees		
			2.3.9.2.3	Support Vector Machines		
	2.3.10	Compare Machine Learning Algorithms				02 Hours
	2.3.11	Automate Machine Learning Workflows with Pipelines				
		2.3.11.1	Data Preparation and Modeling Pipeline			
		2.3.11.2	Feature Extraction and Modeling Pipeline			
	2.3.12	Performance Improvement with Ensembles				02 Hours
		2.3.12.1	Combine Models Into Ensemble Predictions			
			2.3.12.1.1	Bagging Algorithms		
				2.3.12.1.1.1	Concepts of Begging	

					Algorithm	
				2.3.12.1.1.2	Random Forest	
			2.3.12.1.2	Boosting Algorithms		
				2.3.12.1.2.1	Concepts of Boosting Algorithm	
				2.3.12.1.2.2	AdaBoost	
			2.3.12.1.3	Voting Ensemble		
	2.3.13	Performance Improvement with Algorithm Tuning				02 Hours
		2.3.13.1	Grid Search Parameter Tuning			
		2.3.13.2	Random Search Parameter Tuning			

Module 03: Concepts and Uses of Deep Learning and Real Life Projects

Section	Topics		Duration
3.1	**Basic of Biological Neural Networks and It’s Learning Process**		
	3.1.1	Structure of Biological Neuron	02 hours
	3.1.2	Learning Process of Biological Neuron	
3.2	**Pattern recognition, Feature vector and Feature Vector**		02 Hours
	3.2.1	Definition of Pattern Recognition	
	3.2.2	Block Diagram of Typical Pattern Recognition System	
	3.2.3	Feature Vector and Feature Space with Real Life Example	
3.3	**Simulation Procedure of Biological Neuron into Machine**		02 Hours
	3.3.1	Outline of Basic Model of Biological Neuron	
	3.3.2	Overview of Thresholding Function	
	3.3.3	Single Layer Perceptron Learning Algorithm	
	3.3.4	Limitations of Single Layer Perceptron Learning Algorithm	
3.4	**Multilayer Perceptron Neural Networks Algorithms**		04 Hours
	3.4.1	Architectural Overview of Multilayer Perceptron Neural Networks	
	3.4.2	Credit Assignment Problem, Characteristics of Linear and Sigmoidal Thresholding Function	
	3.4.3	Algorithm Steps of Multilayer Perceptron Neural Networks Algorithm	
	3.4.4	How does Hidden Layer Act as a Feature Detector?	
	3.4.5	Visualizing Network Behaviour in 2D and 3D Form	
	3.4.6	Decision Boundaries or Regions Formed by introducing Different Number of Layers	
3.5	**Deep Learning with Python**		

3.5.1	Deep Learning Libraries in Python		
	3.5.1.1	Introduction to Keras, Theano and TensorFlow Backends for Keras	02 Hours
	3.5.1.2	Build Deep Learning Models with Keras	
3.5.2	Develop the First Neural Network With Keras		
	3.5.2.1	Load Data	
	3.5.2.2	Define Model	
	3.5.2.3	Compile Model	
	3.5.2.4	Fit Model	
	3.5.2.5	Evaluate Model	
3.5.3	Evaluate The Performance of Deep Learning Models		02 Hours
	3.5.3.1	Empirically Evaluate Network Configurations	
	3.5.3.2	Data Splitting	
	3.5.3.3	Manual k-Fold Cross Validation	
3.5.4	Use Keras Models With Scikit-Learn For General Machine Learning		
	3.5.4.1	Evaluate Models with Cross Validation	
	3.5.4.2	Grid Search Deep Learning Model Parameters	
3.5.5	Project: Multiclass Classification Of Flower Species		02 Hours
	3.5.5.1	Overview of Iris Flowers Classification Dataset	
	3.5.5.2	Import Classes and Functions	
	3.5.5.3	Initialize Random Number Generator	
	3.5.5.4	Load The Dataset	

	3.5.5.5	Encode The Output Variable	
	3.5.5.6	Define The Neural Network Model	
	3.5.5.7	Evaluate The Model with k-Fold Cross Validation	
3.5.6	Project: Binary Classification Of Sonar Returns		02 Hours
	3.5.6.1	Overview of Sonar Object Classification Dataset	
	3.5.6.2	Baseline Neural Network Model Performance	
	3.5.6.3	Improve Performance With Data Preparation	
	3.5.6.4	Tuning Layers and Neurons in The Model	
3.5.7	Project: Regression Of Boston House Prices		02 hours
	3.5.7.1	Overview of Boston House Price Dataset	
	3.5.7.2	Develop a Baseline Neural Network Model	
	3.5.7.3	Lift Performance By Standardizing The Dataset	
	3.5.7.4	Tune The Neural Network Topology	
3.5.8	Understand Model Behavior During Training By Plotting History		02 Hours
	3.5.8.1	Access Model Training History in Keras	
	3.5.8.2	Visualize Model Training History in Keras	
3.5.9	Reduce Overfitting With Dropout Regularization		
	3.5.9.1	Dropout Regularization For Neural Networks	
	3.5.9.2	Dropout Regularization in Keras	
	3.5.9.3	Using Dropout on the Visible Layer	
	3.5.9.4	Using Dropout on Hidden Layers	

3.5.10	Lift Performance With Learning Rate Schedules		02 Hours
	3.5.10.1	Learning Rate Schedule For Training Models	
	3.5.10.2	Ionosphere Classification Dataset	
	3.5.10.3	Time-Based Learning Rate Schedule	
	3.5.10.4	Drop-Based Learning Rate Schedule	
3.5.11	Basic Concepts of Convolutional Neural Networks		02 Hours
	3.5.11.1	The Case for Convolutional Neural Networks	
	3.5.11.2	Building Blocks of Convolutional Neural Networks	
	3.5.11.3	Convolutional Layers	
	3.5.11.4	Pooling Layers	
	3.5.11.5	Fully Connected Layers	
	3.5.11.6	Worked Example	
	3.5.11.7	Convolutional Neural Networks Best Practices	
3.5.12	Project: Handwritten Digit Recognition		02 Hours
	3.5.12.1	Overview of Handwritten Digit Recognition Dataset	
	3.5.12.2	Loading the MNIST dataset in Keras	
	3.5.12.3	Baseline Model with Multilayer Perceptrons	
	3.5.12.4	Simple Convolutional Neural Network for MNIST	
	3.5.12.5	Larger Convolutional Neural Network for MNIST	
3.5.13	Improve Model Performance With Image Augmentation		02 Hours
	3.5.13.1	Keras Image Augmentation API	

		3.5.13.2	Point of Comparison for Image Augmentation	
		3.5.13.3	Feature Standardization	
		3.5.13.4	ZCA Whitening	
		3.5.13.5	Random Rotations	
		3.5.13.6	Random Shifts	
		3.5.14.7	Random Flips	
		3.5.14.8	Saving Augmented Images to File	
	3.5.14	Project Object Recognition in Photographs		02 Hours
		3.5.14.1	Overview of Photograph Object Recognition Dataset	
		3.5.14.2	Loading The CIFAR-10 Dataset in Keras	
		3.5.14.3	Simple CNN for CIFAR-10	
		3.5.14.4	Larger CNN for CIFAR-10	
		3.5.14.5	Extensions To Improve Model Performance	
	3.5.15	Project: Predict Sentiment From Movie Reviews		02 Hours
		3.5.15.1	Movie Review Sentiment Classification Dataset	
		3.5.15.2	Load the IMDB Dataset With Keras	
		3.5.15.3	Word Embeddings	
		3.5.15.4	Simple Multilayer Perceptron Model	
		3.5.15.5	One-Dimensional Convolutional Neural Network	

Final Assessment Examination	02 Hours