DEPARTMENT OF COMPUTER SCIENCE

MSc in Data Science and Artificial Intelligence

Programme Syllabus

Ramakrishna Mission Vivekananda Educational and Research Institute

Belur Campus

MSc in Big Data Analytics
Syllabus

Contents

Programme Outcomes

1. Inculcate critical thinking to carry out scientific investigation objectively without being biased with preconceived notions.
2. Equip the student with skills to analyze problems, formulate an hypothesis, evaluate and validate results, and draw reasonable conclusions thereof.
3. Prepare students for pursuing research or careers in industry in mathematical sciences and allied fields
4. Imbibe effective scientific and/or technical communication in both oral and writing.
5. Learn the necessary tools and programming skills needed for data analysis and visualisation
6. Continue to acquire relevant knowledge and skills appropriate to professional activities and demonstrate highest standards of ethics in learning and work

Programme Specific Outcomes

1. Basic understanding of statistical methods, probability, mathematical foundations, and computing methods relevant to data analytics.
2. Knowledge about storage, organization, and manipulation of structured data.
3. Understand the challenges associated with big data computing.
4. Training in contemporary big data technologies
5. Understanding about the data analytics pipeline beginning with problem identification and translation, followed by model building and validation with the aim of knowledge discovery in the given domain.
6. Ability to write good project reports and do effective presentations on technical topics

Artificial Intelligence

Course Code: CS246

Course Description: Artificial Intelligence (AI) is a vast field. Nowadays people often think that AI is nothing but machine Learning and the reason is mainly the widespread use of machine learning. The fact is machine learning is just a part of the domain of AI. We will try to explore the full breadth of the course, which encompasses logic, probability, reasoning, learning, decision making, and action.
We will define AI as the study of intelligent agents where each such agent receives precept from the environment and act upon the precept. We will discuss goal-based models, knowledge-based models,
After a basic introduction, we will jump into different modules like problem-solving (for example, Sudoku solver), planning, and acting under uncertainty. Application of artificial intelligence in various fields like natural language processing, perception (computer vision), robotics will be covered.

Prerequisite(s): Basic knowledge of computer science such as algorithms, data structures, probability, linear algebra.
Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Credit Hours: 4

Text(s):

Artificial Intelligence: A Modern Approach ;
Stuart Russell and Peter Norvig

Course Objective: Students will get to know

(1) Understand the fundamental concepts of Artificial Intelligence (AI), including intelligent agents, environments, and the broad scope of AI beyond machine learning

(2) Learn various problem-solving strategies such as searching, informed exploration, and constraint satisfaction, and apply them to solve complex problems

(3) Explore planning techniques and decision-making processes in the presence of uncertainty, including probabilistic reasoning and making both simple and complex decisions

(4) Gain an overview of learning methodologies, including supervised, unsupervised, and reinforcement learning, and understand their applications in AI problem-solving

(5) Understand the principles of Bayesian Networks and Hidden Markov Models (HMMs), including their representations, inference methods, and applications in real-world scenarios
(6) Explore the diverse applications of AI in fields such as robotics, computer vision, natural language processing, and computer games, including deep reinforcement learning techniques

Course Outcome: The students will be able to
(1) Analyze and identify problems suitable for the application of Artificial Intelligence techniques across various domains, including but not limited to problem-solving, planning, and decision-making
(2) Implement selected AI techniques to effectively address and solve complex problems, demonstrating proficiency in foundational AI concepts such as problem-solving by searching, constraint satisfaction, and adversarial search
(3) Evaluate and apply state-of-the-art AI techniques, including probabilistic reasoning, reinforcement learning, and Bayesian Networks, to address real-world challenges in diverse fields such as robotics, computer vision, and natural language processing
(4) Demonstrate a comprehensive understanding of key AI principles and methodologies, as outlined in the course syllabus, including the principles of intelligent agents, uncertain knowledge and reasoning, and learning algorithms.
(5) Critically assess the applicability of basic and advanced AI techniques to different problem domains, making informed decisions about selecting and implementing appropriate methods based on problem characteristics and constraints.
(6) Through a combination of theoretical understanding, practical application, and critical analysis, students will develop the necessary skills and knowledge to contribute effectively to the field of Artificial Intelligence and apply AI techniques to solve complex real-world problems.

Grade Distribution:

Assignments and Class Tests	30%
Midterm Exam	20%
Final Exam	50%

Course Outline (tentative) and Syllabus:

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures. Quizzes will be unannounced.

Week	Topic
1	Introduction, Intelligent Agents and Environments
2	Problem Solving: Solving problem by searching
3	Problem Solving: Informed search and exploration
4	Problem Solving: Constraint satisfaction problem
5	Problem Solving: Adversarial Search
6	Uncertain Knowledge and Reasoning: Uncertainity
7	Uncertain Knowledge and Reasoning: Probabilistinc reasoning
8	Uncertain Knowledge and Reasoning: Making simple decisions
9	Uncertain Knowledge and Reasoning: Making complex decisions
10	Learning: Brief overview of various learning methods
11	Learning: Reinforcement learning
12	Bayes Net (1)
13	Bayes Net (2)
14	Application: Computer vision, Natural Language Processing
15	Deep Reinforcement Learning
16	Course Wrap-up

Applied Machine Learning (reading course)

Course Code: DA342

Course Description: DA342 will explore some Machine Learning (ML) techniques to solve different practical problems that occur in one of the area of *Audio, Text, Image, and Video processing* task. We will start with the definition of the particular problem and some initial results. Then try to touch some state-of-the-art results from the papers published in the last *ten* years in the top venues like ICML, NeurIPS, CVPR, ICCV, ECCV and related journals.

Prerequisite(s): Student should have some knowledge in

- Mathematics: Linear Algebra, Multivariate Calculus, Basic Optimization, Basic Probability and Statistics
- Computer programming: Any one from C/C++/Python(recommended for the class project)/MATLAB/Octave
- Basis ML concepts (classification, regression, etc.)

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Credit Hours: 4

Text(s):

Probabilistic Machine Learning: An Introduction;
Kevin Patrick Murphy

Pattern Recognition and Machine Learning;
Christopher M. Bishop

Statistical Learning with Sparsity: The Lasso and Generalizations;
Trevor Hastie, Robert Tibshirani and Martin Wainwright

Pattern Classification and Scene Analysis;
R. O. Duda, P. E. Hart and D. G. Stork

Foundations of Machine Learning;
Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar

Understanding Machine Learning: From Theory to Algorithms;
Shai Shalev-Shwartz and Shai Ben-David

Speech and Language Processing;
Dan Jurafsky and James H. Martin.

Deep Learning;
Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar

Course Objective: Students will get to know

(1) critical reading and analysis skills for different practical applications of machine learning algorithms

(2) identifying main concepts, key arguments, evaluating evidence, and understanding the author's perspective and biases

(3) recognizing the limitations of machine learning algorithms in addressing real-world problems

Course outcomes: After successful completion of this course, students will be able to:

(1) decompose a data related problem (in the area of *Audio, Text, Image, and Video processing*) into subproblems in ML and identify potential solutions

(2) apply state-of-the-art ML tools and techniques to solve data related problems that occur in one of the area of *Audio, Text, Image, and Video processing* task

(3) start critically read, assess, and synthesize scientific literature in applied ML and related fields

(4) discuss the potential and limitation of ML models in a particular area

Grade Distribution:

Paper reading and presentation	40%
Project and report writing	50%
Paper/Project review	10%

Course Outline (tentative) and Syllabus:

There is no predefined(fixed) syllabus for this course! The instructor will give some topics based on the current hot research topic and industry problems with a brief introduction. Students have to choose one topic based on their interests. The instructor will share some materials related to the selected topics. Each student has

to present those at least once a week and have to do a project on that or a related topic.
In 2022, initially, four students opted for this course, and two have left! They have selected the following topics:

- Student-1: **Fake news/misinformation detection.**
- Student-2: **Tackling climate change with machine learning.**
- Student-3 (left): **Player/team performance prediction in IPL game.**
- Student-4 (left): **Sparse coding/learning and its application in images and video data.**

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures. Reading assignments are from reference textbook will be mentioned in the suggested reading material in the *course url*.

Week	Content
Week 1	• Introduction to the course • Course logistics • A brief discussion of some practical problems (research and industry)
Week 2	• A brief discussion of more practical problems (research and industry) and asked the student to explore some focused workshops happening in the top venues like ICML, NeurIPS, CVPR, ICCV, ECCV, ICLR etc.. • Assignment of 2-3 topics to each student.
Week 3	• Particular topic selection for each student – Student has to give a brief presentation on the topics allocated in the last week
Week 4	• Student presentation
Week 5	• Student presentation
Week 6	• Student presentation • Finalise the class project (through a public presentation)
Week 7	• Student presentation and project update
Week 8	• Student presentation and project update
Week 9	• Student presentation and project update
Week 10	• Student presentation and project update
Week 11	• Student presentation and project update
Week 12	• Student presentation and project update

Week 13	• Student presentation and project update – Discussion on project report
Week 14	• Student presentation and project update – Review of the project report
Week 15	• Final presentations (public)

Basic Statistics

Course Code: DA102

Course Description: DA102 is going to provide an introduction to some basic statistical methods for analysis of categorical and continuous data. Students will also learn to make practical use of the statistical computer package R.

Prerequisite(s): NA
Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.
Credit Hours: 4

Text(s):

Statistics;
David Freedman, Pobert Pisani and Roger Purves

Fundamentals of Statistics (V-1 & 2),
A.M. Goon, M. K.Gupta, B. Dasgupta

The visual display of Quantitative Information;
Edward Tufte

Mathematical Statistics with Applications;
Kandethody M. Ramachandran and Chris P. Tsokos

Fundamental of Mathematical Statistics
S. C. Gupta and V. K. Kapoor

Course Objectives: Students will get to know

(1) fundamental statistical concepts and some of their basic applications in real world.
(2) organizing, managing, and presenting data,
(3) how to use a wide variety of specific statistical methods, and,
(4) computer programming in R.

Course Outcome: The students will be able to

(1) apply technologies in organizing different types of data,
(2) present results effectively by making appropriate displays, summaries, and tables of data,
(3) perform simple statistical analyses using R

(4) analyze the data and come up with correct interpretations and relevant conclusions.

Grade Distribution:

Assignments	20%
Quizzes	10%
Midterm Exam	20%
Final Exam	50%

Course Outline (tentative) and Syllabus:

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures. Quizzes will be unannounced.

Week	Content
Week 1	• Introduction: What is Statistics • Types of Data: – Quantity/Qualitative – Identity – Freq/Non-freq • Data Collection: Direct observation, Direct question, Questionnaire
Week 2	• Data Representation: Textual, Tabular • Series Data and Grouped Data • Graphical Representation: – Bar diagram – Pie-chart – Frequency bar chart – Histogram – Ogive
Week 3	• Descriptive Numerical Measures – Measures of Central Tendency: AM, GM, HM, Median, Quantile, Mode – Box Plot
Week 4	• Descriptive Numerical Measures – Measures of Dispersion: Range, IQR, Mean Deviation, Root mean square deviation (SD, Variance), Gini Coefficient, CV, CQD. – Measure of Skewness – Measure of Kurtosis
Week 5	• Problem Session • Quiz 1
Week 6	• Bivariate Data – Scatter plot • Correlation – Pearson's correlation, Spearman's correlation, Kendall's correlation – Correlation with ties
Week 7	• Method of Curve Fitting – Method of least square

Week	Content
Week 9	• Normal Distributions and its associates – χ^2 distribution – t distribution – F distribution
Week 10	• Statistical Inference • Estimation – Point Estimation – Interval Estimation
Week 11	• ForestGreenMethods of calculating point estimates – Method of moments – Method of maximum likelihood
Week 12	• ForestGreenProperties of Point Estimators: Unbiased – Bias – MSE – Minimum Variance Unbiased Estimator
Week 13	• ForestGreenProperties of Point Estimators: Sufficiency – Neyman–Fisher factorization theorem – Joint Sufficiency – RAO–BLACKWELL Theorem
Week 14	• ForestGreenProperties of Point Estimators: Sufficiency – Minimal sufficient statistic – Lehmann and Scheffe method • Properties of Point Estimators: Consistency – Test for Consistency
Week 15	• ForestGreenProperties of Point Estimators: Efficiency – Uniformly Minimum Variance Unbiased Estimator – Cramer–Rao Lower bound – Fisher Information

Computer Vision and Pattern Recognition

Course Code: CS342

Course Description: In an era where cameras and digital imagery permeate our daily lives, understanding computer vision has become essential. CS342 provides an introductory exploration into this fascinating field. From casual "selfies" to immersive augmented reality experiences, the applications of computer vision are ubiquitous.

In this course, we delve into critical questions that shape the landscape of computer vision today:How do computers efficiently search for specific images amidst vast datasets? What algorithms enable the automatic tagging of photos on social media platforms? Can machines accurately interpret natural gestures or sign languages for seamless human-computer interaction? How do self-driving cars navigate complex terrains using visual input?

Our journey spans a wide range of topics, from the basics of image processing to cutting-edge visual recognition techniques. We explore machine learning methods, including supervised algorithms and deep neural networks, to tackle intricate research problems.

Moreover, we engage in discussions about ethical considerations, privacy, and bias in the responsible development of AI-driven systems.

Prerequisite(s): Basic knowledge of probability and linear algebra; data structures, algorithms; programming experience. Previous experience with image processing will be useful but is not assumed.
Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Credit Hours: 4

Text(s):

Computer Vision: Algorithms and Applications;
Richard Szeliski

Digital Image Processing;
Rafael Gonzalez and Richard Woods

Computer Vision: A Modern Approach;

David A. Forsyth and Jean Ponce

Computer Vision;
Linda G. Shapiro and George C. Stockman

Multiple View Geometry in Computer Vision;
Richard Hartley and Andrew Zisserman

Pattern classification;
Richard O. Duda, Peter E. Hart, and David G. Stork

Pattern Recognition and Machine Learning;
Christopher M. Bishop

Course Objective: Students will get to know

(1) a comprehensive understanding of the theoretical and practical aspects of computing with images, including image formation, measurement, and analysis.

(2) Students will implement common methods for robust image matching and alignment, demonstrating proficiency in image processing techniques.

(3) Students will grasp the geometric relationships between 2D images and the 3D world, enabling them to understand concepts such as image transformations, alignment, and epipolar geometry (4) Students will gain exposure to object and scene recognition and categorization from images, including techniques such as object detection, supervised classification algorithms, and probabilistic models for sequence data

(5) Students will comprehend the principles of state-of-the-art deep neural networks, starting with convolutional neural networks (CNNs), and their applications in computer vision

(6) Students will develop practical skills necessary to build computer vision applications, enabling them to apply learned concepts in real-world scenarios effectively

(7) Students will view Computer Vision as a research area, understanding its ongoing developments, challenges, and potential applications

Course Outcome: The students will be able to

(1) demonstrate proficiency in image processing techniques, including feature extraction, filtering and edge detection through practical implementation and application exercises.

(2) Students will acquire the ability to employ segmentation and clustering algorithms, Hough transform, and robust fitting techniques, such as RANSAC for grouping and fitting objects and contours within images.

(3) Students will develop a deep understanding of multiple view geometry, local invariant feature detection and description, image transformations, planar homography, and stereo vision techniques for analyzing images and objects from different perspectives. (4) Students will demonstrate competence in object and scene recognition, categorization, and classification using supervised learning algorithms, probabilistic models, support vector machines, and neural networks
(5) Students will gain practical experience in building computer vision applications, utilizing learned techniques to address real-world problems, such as object detection and image classification
(6) Through assignments, projects, and exams, students will develop critical thinking skills and problem-solving abilities necessary for tackling challenges in computer vision and pattern recognition domains
(7) Students will develop a research-oriented perspective towards computer vision, gaining exposure to advanced topics, current trends, and emerging techniques in the field through lectures, readings, and discussions
(8) Students will enhance their communication skills through project and paper presentations
(9) Students will explore ethical considerations and implications related to computer vision technologies, understanding issues such as privacy, bias, and societal impact, and developing a responsible approach towards the use and development of computer vision systems
(10) By the end of the course, students will recognize the importance of lifelong learning in the rapidly evolving field of computer vision, being prepared to adapt to new technologies, methods, and research findings throughout their careers

Grade Distribution:

Assignments and Class Tests	30%
Midterm Exam	20%
Final Exam	50%

Course Outline (tentative) and Syllabus:

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures. Quizzes will be unannounced.

Week	Topic
1	Introduction and Basics of Digital Image
2	Filtering, Edge Detection and Template Matching
3	Frequency Response and Hough Transform
4	Harris Corner, SIFT
5	Subsampling, Fitting Interpolation
6	Tracking, Binary Image
7	Camera Stereo, Multiview Geometry
8	Camera Parameters, Panorama, Disparity
9	Machine Learning in Computer Vision
10	Machine Learning to Deep Learning
11	CNN, RNN, LSTM
12	Transformers, GNN
13	Segmentation
14	Generative AI for Image Generation
15	Paper Presentations
16	Advanced topics (if time permits), Applications, Course Wrap-up

Machine Learning with Graphs

Course Code: CS410

Course Information

Credits: 2 (30 hrs)
Prerequisites: Calculus, Linear Algebra, Probability and Statistics, Machine Learning, Algorithms, Graph Theory, Python Programming.

Course Description

Complex data can be represented as a graph of relationships and interactions between objects. This course explores the computational, algorithmic, and modeling challenges specific to the analysis of large-scale graphs. By examining graph structures, students will learn machine learning and data mining techniques to improve predictions and gain insights into various networked systems.

Course Goals and Objectives

At the end of this course, students will be able to:

- Understand and apply **Graph Neural Networks (GNNs)**.
- Perform **representation learning** and generate **node embeddings**.
- Conduct **link analysis** and study the **graph structure of the web**.
- Explore **models of network evolution** and network cascades.
- Reason over **knowledge graphs**.
- Use **deep generative models** for graph data.
- Detect communities and clusters within networks.

Weekly Schedule (Tentative)

1. **Week 1:** Introduction; Structure of Graphs
2. **Week 2:** Traditional Machine Learning on Graphs
3. **Week 3:** Node embeddings
4. **Week 4:** Introduction to Graph Neural Networks (GNNs)

5. **Week 5:** A general perspective on GNNs
6. **Week 6:** GNN augmentation and training
7. **Week 7:** Theory of GNNs
8. **Week 8:** Heterogeneous graphs
9. **Week 9:** Knowledge graphs
10. **Week 10:** Reasoning over knowledge graphs
11. **Week 11:** Fast neural subgraph matching
12. **Week 12:** GNNs for recommenders
13. **Week 13:** Deep generative models for graphs
14. **Week 14:** Advanced topics in GNNs
15. **Week 15:** Scaling to large graphs

Reference Books

1. *Graph Machine Learning* by Claudio Stamile, Aldo Marzullo, and Enrico Deusebio
2. *Introduction to Graph Neural Networks* by Zhiyuan Liu, Jie Zhou
3. *Graph Representation Learning* by William L. Hamilton
4. *Deep Learning on Graphs* by Yao Ma, Jilian Tang
5. *Graph Neural Networks: Foundations, Frontiers, and Applications* by Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao

Deep Learning (DL) and its Application in Natural Language Processing (NLP)

Course Code: **DA345**

Course description:

This course presents an introduction to the **Deep Learning (DL)** and its application in **Natural Language Processing (NLP)**. We will explore theoretical foundation of DL and their practical implementation to solve different NLP related tasks such as *text classification*, *sentiment analysis*, *machine translation*, *text summarization*, *text generation*, etc..

Prerequisite (s):

Student should have some knowledge in

- Mathematics: *Linear Algebra, Multivariate Calculus, Basis Optimisation and Basic probability*
- Computer programming: *Any one from C/C++/Python (recommended for the class project and assignments)/MATLAB/Octave*
- Basic concept in Algorithms and Data Structure
- Introduction to Machine Learning

Credit Hours: 4, approximately 60 credit hours

Text(s):

- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.*, MIT Press, 1st edition, 2016. online
- Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*, Cambridge University Press, 2023. online
- Simon J.D. Prince. *Understanding Deep Learning*, MIT Press, 2023. online
- Eugene Charniak. *Introduction to deep learning*, MIT Press, 2018.
- Michael Nielsen. *Neural Networks and Deep Learning*, online

- Dan Jurafsky and James H. Martin. *Speech and Language Processing*. draft, 3rd edition, 2024. [online]
- Delip Rao, Brian McMahan. *Natural Language Processing with PyTorch*, O'Reilly Media, Inc, 2019
- Lewis Tunstall, Leandro von Werra, and Thomas Wolf. *Natural Language Processing with Transformers Book*, O'Reilly Media, Inc, 2022. online code only
- Yoav Goldberg. *A Primer on Neural Network Models for Natural Language Processing*. online

Course Objective:

Students will get to know:

- the core concepts of DL, including network design, training and evaluation.
- common DL architechtures for tasks like object recognition, image generation, sentiment analysis, text summarization, and machine translation, etc..
- techniques for text preprocessing to clean and prepare textual data for NLP tasks.
- advanced deep learning models for different text processing tasks in NLP
- use of different DL and NLP toolkits (like PyTorch, NLTK, PyTorch) to solve practical NLP tasks.
- the performance of DL models and identify potential biases or errors.

Course outcomes:

After successful completion of this course, students will be able to:

- understand the mathematical basis of various deep-learning models.
- explain key deep learning concepts such as activation functions, loss functions, backpropagation, regularization, etc..
- design, implement, and evaluate feedforward, convolutional, recurrent neural networks and modern transformer based architechtures.
- explain textual data and extract representative features.
- solve practical problems in NLP using deep learning methods.

- gain hands-on experience with deep learning models in PyTorch [1].

Grade Distribution:

Quizzes and assignments	25%
Midterm Exam	10%
Class project	25%
Final Exam	40%

Course outline (tentative) and syllabus:

The tentative syllabus is as follows:

- Artificial neural network (ANN): Modelling Single neuron activity, different types of activity functions (sigmoid, tanh, ReLU, ELU etc.), how to connect multiple neurons to form a network, Multi-layer perceptron
- Optimization: Back propagation, different loss functions, gradient decent, stochastic gradient decent and different update rules (AdaGrad, RPMSprops, Adam etc.) for network parameters, regularization, dropout, batch normalisation etc.
- Deep learning toolbox: Explore a deep learning toolbox like PyTorch (my personal choice)/ TensorFlow and their autograd functionalities
- Convolutional neural network (CNN): Concept of kernel and convolution, some pooling operation (max, average etc.), some standard CNN architectures like LeNet, AlexNet, VggNet, ResNet etc. and concept of transfer learning
- Recurrent neural network (RNN): Sequential data and how to handle those using neural network, general RNN architecture, some popular RNN architectures like *Long short-term memory (LSTM)*, *Gated recurrent unit (GRU)* and their different variants
- Deep generative models: *Variational Autoencoders (VAE)*, *Generative Adversarial Networks (GAN)*, *Normalizing flows*, *Diffusion models*, etc.
- Neural language model:

[1] https://pytorch.org/

- Introduction to NLP
- Text preprocessing: tokenisation, stop words, stemming, lemmatisation, etc.
- Vector representations of text: *Bag of Words*, *TF-IDF*, *word embeddings*, *Word2Vec*, *GloVE*, etc.
- Sequence modelling: *Recurrent neural network (RNN)*, *Self-Attention network*, etc.
- Transformers: *Attention, BERT* and its different variants, Encoder-Decoder models
- Large language model (LLM): *GPT* different variants, *pre-trained language model*, *transfer learning*
- Application: *text classification*, *sentiment analysis*, *Named Entity Recognition (NER)*, *machine translation*, *text summarization*, *text generation*, etc.

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures. Reading assignments are from reference textbook will be mentioned in the suggested reading material in the *course url*.

Week	Content
Week 1	• Motivation: Artificial neural network (ANN): Modelling Single neuron activity, different types of activity functions (sigmoid, tanh, ReLU, ELU etc.), how to connect multiple neurons to form a network, Multi-layer perceptron
Week 2	• Optimization: Back propagation, different loss functions, gradient decent, stochastic gradient decent and different update rules (AdaGrad, RPMSprops, Adam etc.) for network parameters, regularization, dropout, batch normalisation etc. • Assignment-1
Week 3	• Deep learning toolbox: Explore a deep learning toolbox like PyTorch (my personal choice)/ TensorFlow and their autograd functionalities • Assignment-2
Week 4	• Convolutional neural network (CNN): Concept of kernel and convolution, some pooling operation (max, average etc.), some standard CNN architectures like LeNet, AlexNet, VggNet, ResNet etc. and concept of transfer learning • Assignment-3 • Class project discussion
Week 5	• Recurrent neural network (RNN): Sequential data and how to handle those using neural network, general RNN architecture, some popular RNN architectures like *Long short-term memory (LSTM)*, *Gated recurrent unit (GRU)* and their different variants • Assignment-4
Week 6	• Class project proposal • Deep generative models: *Variational Autoencoders (VAE)* • Assignment-5
Week 7	• Deep generative models: *Generative Adversarial Networks (GAN)*, *Normalizing flows*, *Diffusion models*, etc. • Course review for the mid-term examination
Week 8 - 15	• Neural language model: – Introduction to NLP – Text preprocessing: tokenisation, stop words, stemming, lemmatisation, etc. – Vector representations of text: *Bag of Words*, *TF-IDF*, *word embeddings*, *Word2Vec*, *GloVE*, etc. – Sequence modelling: *Recurrent neural network (RNN)*, *Self-Attention network*, etc. – Transformers: *Attention, BERT* and its different variants, Encoder-Decoder models – Large language model (LLM): *GPT* different variants, *pre-trained language model*, *transfer learning*

Data Structures and Algorithms

Course Code: CS110

Course Description:

The course delves into the fundamental components that are needed for efficient and effective programming: data structures and algorithms. It covers essential data structures such as arrays, linked lists, trees, and graphs, highlighting their unique strengths, limitations and applications where they are commonly used. The course discusses essential algorithms for tasks such as searching, sorting, and traversing. It also examines techniques to analyse the time and space complexity of algorithms. Throughout the course, there will be extensive implementation of these algorithms in Python, not only enhancing students' programming skills for solving real-world problems but also sharpening their analytical and critical thinking abilities.

Prerequisite(s): Basic Programming Skills in C/C++/ Python

Note: Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. Students, however, will be evaluated only on the basis of topics covered in the course.

Course Objectives:

- Learn programming in Python
- Analyse the performance of algorithms
- Introduce fundamental data structures
- Explain the characteristics and trade-offs of different data structures
- Develop proficiency in designing and implementing various types of algorithms
- Apply data structures and algorithms to solve real-world problems
- Promote problem-solving skills: Analytical thinking, logical reasoning, algorithmic design.

Course Outcomes:

- Able to programme efficiently in Python.
- Analyze the time and space complexity of algorithms using Big O, Omega, Theta notations.
- Demonstrate understanding of different data structures and their operations.

- Choose the appropriate data structure for a given problem based on its characteristics and constraints.
- Implement common data structures and algorithms efficiently in chosen programming language (Python).
- Design and develop own algorithms for specific tasks.
- Improve problem-solving skills through algorithmic thinking and reasoning.
- Communicate algorithms and solutions effectively in written and verbal forms.

Approximate weightage of different components in evaluation:

Assignments/Tests	20%
Midterm Exam	30%
Final Exam	50%

Course Policies:

General

1. Computing devices are not to be used during any exams unless instructed to do so.
2. Quizzes and exams are closed books and closed notes.
3. Quizzes are unannounced but they are frequently held after a topic has been covered.
4. ***No makeup quizzes or exams will be given.***

Attendance and Absences

1. Attendance is expected and will be taken in each class. Students are not supposed to miss class without prior notice/permission. Any absences may result in point and/or grade deductions.
2. Students are responsible for all missed work, regardless of the reason for absence. It is also the absentee's responsibility to get all missing notes or materials.

Textbooks(s):

1. Data Structures and Algorithms in Python (An Indian Adaptation)
   Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser
2. Introduction to Algorithms, 4^{th} edition
   Thomas H. Cormen, Charles E. Leiserson, Ronald Rivest, Clifford Stein
3. Algorithms, 4^{th} Edition
   Robert Sedgewick and Kevin Wayne

Course Outline (tentative) and Syllabus:

The weekly coverage might change as it depends on the progress of the class. Each week assumes 4 hour lectures and 2 hour practicals (as and when required). Quizzes will be unannounced, so students should maintain close to 100% attendance.

Week	Content
Week 1 - 3	• Introduction to Python programming • Fundamentals of object-oriented programming in Python
Week 4	• Introduction to DSA • Algorithm Analysis, Asymptotic Analysis - O, Ω, θ, o, ω
Week 5	• Problem solving session • Recursion, Analyzing Recursive Algorithms, Tower of Hanoi, Recursion types, tail recursion, • Introduction of the Divide and Conquer paradigm of algorithm design: merge sort, binary search, Master theorem
Week 6	• Array-Based Sequences, Python's Sequence Types • Dynamic Arrays and Amortization, Efficiency of Python's Sequence Types • Multidimensional data
Week 7	• Stacks • Queues, and Deques
Week 8	• Quiz 1 • Linked Lists • Review for Midterm exam
Week 9	• Trees, Traversal Algorithms
Week 10	• Priority Queues • Heaps
Week 11	• Maps, Hash Tables
Week 12	• Sorting: Quick sort, Selection, Insertion, Randomized Quick-Select, • Comparision of sorting algorithms
Week 13	• Search Trees - Binary Search, AVL, and Red-Black Trees
Week 14	• Graph Algorithms - Traversals, Shortest path, Minimum Spanning Trees
Week 15	• Introducing the concept of Dynamic Programming and use of memoization • Greedy Methods

Introduction to Econometrics

Course Code: DA240

Course Description:

This course is going to provide a broad introduction to the most fundamental methodologies and techniques used in Econometrics. Students will learn the details of regression analysis and its applications in real life scenario.
Prerequisite(s): Basic Statistics and probability

Note: Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. Students, however, will be evaluated only on the basis of topics covered in the course.

Course Objectives:

- Understand linear regression assumptions and their implications.
- Detect and address multicollinearity, heteroscedasticity, and endogeneity.
- Apply least squares estimation, generalized least squares, and instrumental variables.
- Interpret model results and assess model validity.

Course Outcomes:

- Fundamental understanding: Students will have a strong grasp of linear regression concepts, assumptions, and limitations.
- Issue identification: Students can detect and address issues like multicollinearity, heteroscedasticity, and endogeneity.
- Estimation techniques: Students can apply least squares, generalized least squares, and instrumental variables.
- Model interpretation: Students can interpret coefficients, goodness of fit, and diagnostic statistics.
- Model validation: Students can assess model validity through assumptions and diagnostic tests.
- Practical application: Students can use linear regression in real-world data analysis across various fields.

Approximate weightage of different components in evaluation:

Assignments/Tests	20%
Midterm Exam	30%
Final Exam	50%

Course Policies:

General

1. Computing devices are not to be used during any exams unless instructed to do so.
2. Quizzes and exams are closed books and closed notes.
3. Quizzes are unannounced but they are frequently held after a topic has been covered.
4. ***No makeup quizzes or exams will be given.***

Attendance and Absences

1. Attendance is expected and will be taken in each class. Students are not supposed to miss class without prior notice/permission. Any absences may result in point and/or grade deductions.
2. Students are responsible for all missed work, regardless of the reason for absence. It is also the absentee's responsibility to get all missing notes or materials.

Textbooks(s):

1. Introduction to Econometrics by G. S. MADDALA.

Course Outline (tentative) and Syllabus:

The weekly coverage might change as it depends on the progress of the class. Each week assumes 2 hour lectures and practicals (as and when required). Quizzes will be unannounced, so students should maintain close to 100% attendance.

Week	Content
Week 1 - 2	• Brief discussion about regression analysis.
Week 3 - 4	• Least Square Estimators
Week 5 - 6	• Multicolinearity
Week 7 - 8	• Heteroscedasticity
Week 9 - 10	• Generalized Least Square Estimation.
Week 11 - 13	• Exogeneity and Endogeneity.
Week 14 - 15	• IV estimator(Instrumental Variable)

Introduction to Finance

Course Code: DA241

Course Description:

This introductory course provides a comprehensive overview of the fundamental concepts and principles of finance. Students will explore the core areas of finance, including corporate finance, financial markets, and investments. The course will delve into topics such as financial statements, time value of money, risk and return, portfolio theory, and capital budgeting. Through case studies and practical exercises, students will gain a solid understanding of how financial decisions are made and their impact on businesses and individuals.
Prerequisite(s): Basic Statistics, probability and stochastic processes.

Note: Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. Students, however, will be evaluated only on the basis of topics covered in the course.

Course Objectives:

- Optimize portfolios: Maximize returns, minimize risk using historical Sensex data.
- Analyze patterns: Identify trends, correlations in company returns.
- Predict returns: Forecast future returns and evaluate accuracy.
- Apply Binomial Model: Use for put-call parity problems and simulations.
- Apply Black-Scholes Model: Use for real-world scenarios and simulations.

Course Outcomes:

- The student will be able to Optimize portfolio on the collected historical Sensex data of different company for giving maximum return with minimum risk.
- Analyze the pattern of return of different company from historical Sensex data.
- Predict the return for a certain amount of time for different company and to check their prediction accuracy from the actual data.
- Apply Binomial Model in real life Put Call parity problems and also understand model working procedure by simulated data.
- Apply Black Sholes formula in real life scenarios and also on simulated data

Approximate weightage of different components in evaluation:

Assignments/Tests	20%
Midterm Exam	30%
Final Exam	50%

Course Policies:

General

1. Computing devices are not to be used during any exams unless instructed to do so.
2. Quizzes and exams are closed books and closed notes.
3. Quizzes are unannounced but they are frequently held after a topic has been covered.
4. ***No makeup quizzes or exams will be given.***

Attendance and Absences

1. Attendance is expected and will be taken in each class. Students are not supposed to miss class without prior notice/permission. Any absences may result in point and/or grade deductions.
2. Students are responsible for all missed work, regardless of the reason for absence. It is also the absentee's responsibility to get all missing notes or materials.

Textbooks(s):

1. John C.Hull- Options, Futures and Other Derivatives
2. Sheldon M. Ross- An elementary introduction to mathematical finance
3. Chi-fu Huang, Robert H. Litzenberger- Foundations for financial economics
4. Gopinath Kallianpur, Rajeeva L. Karandikar- Introduction to option pricing theory

Course Outline (tentative) and Syllabus:

The weekly coverage might change as it depends on the progress of the class. Each week assumes 2 hour lectures and practicals (as and when required). Quizzes will be unannounced, so students should maintain close to 100% attendance.

1. Mortgages and loans and other borrowings. Investment portfolio, portfolio optimization, Different kind of portfolios.

2. Concept of options, Assets , Stocks , Derivatives, Put and Call options (American and European),

3. Arbitrage and Hedging, Uses of them in market scenario

4. Binomial model, Cox-Ross-Rubinstein formula, Black-Sholes formula and their derivation

Week	**Content**
Week 1 - 4	• Mortgages and loans and other borrowings. Investment portfolio, portfolio optimization, Different kind of portfolios.
Week 5 - 8	• Concept of options, Assets , Stocks , Derivatives, Put and Call options (American and European),
Week 9 - 12	• Arbitrage and Hedging, Uses of them in market scenario
Week 13 - 16	• Binomial model, Cox-Ross-Rubinstein formula, Black-Sholes formula and their derivation

Linear Algebra and Matrix Computation

Course Code: DA109

Course Description: This course introduces fundamental concepts of linear algebra and explores practical matrix computation techniques. We will emphasize both theoretical understanding and practical implementation using computational tools. Through lectures, discussions, and problem-solving activities, students will gain a solid understanding of vector spaces, matrices, linear transformations, eigenvalues, and eigenvectors, matrix factorizations, as well as numerical methods for solving linear systems and eigenvalue problems.

Prerequisite(s): Student should have some knowledge in

- Basic concept on algebra, vectors and co-ordinate geometry
- Computer programming: Any one from C/C++/ Python(recommended for the class assignments)/MATLAB/Octave

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Credit Hours: 4

Text(s):

Linear algebra;
A. Ramachandra Rao and P. Bhimasankaram

Linear Algebra Done Right;
Sheldon Axler

Linear Algebra;
Kenneth Hoffman and Ray Kunze

Introduction to linear algebra;
Gilbert Strang

Course Objective: Students will get to know

(1) concepts of vector spaces, including concepts of linear independence, span, and basis
(2) basic operations on vectors and matrices
(3) linear transformations to represent real-world problems

(4) different methods to solve a systems of linear equations
(5) eigenvalues and eigenvectors and their applications
(6) matrix decomposition methods for efficient computations

Course outcomes: After successful completion of this course, students will be able to:
(1) demonstrate the fundamental concepts in linear algebra, including vector spaces, matrices, linear transformations, eigenvalues, and eigenvectors
(2) implement algorithms related to linear algebra and matrix computation using Python programming language
(3) apply matrix computation techniques to solve practical problems, including matrix operations, Gaussian elimination, LU decomposition, matrix inverses, determinants, eigenvalue calculations, and matrix factorizations
(4) analyze and solve linear systems of equations using appropriate methods
(5) apply linear algebra concepts and techniques to solve practical problem in machine learning, data analysis and engineering fields

Grade Distribution:

Assignments and Class Tests	30%
Midterm Exam	20%
Final Exam	50%

Course Outline (tentative) and Syllabus:

The tentative syllabus is as follows:

- Introduction to Vectors: Vectors and their geometry, operations on vectors (addition, multiplication by a scalar, dot product, length).
- Vector Space: Vector space, Subspace, Basis and dimension, Change of basis.
- Linear Transformations: Introduction to linear transformations, Rank-Nullity theorem, Matrix of a linear transformation, Linear operators and isomorphism, Linear functionals.
- Matrix Algebra: Matrix addition and multiplication, transpose, inversion, Special matrices, Row rank and column rank of a matrix, Determinant of a matrix and its geometric interpretation, Cramer's rule to solve system of linear equations, Various matrix decompositions.

- Eigenvalues and Eigenvectors: Introduction to eigenvalues and eigenvectors of matrices, Characteristic polynomial, Cayley-Hamilton theorem, Algebraic and geometric multiplicities of eigenvalues, Matrix diagonalization, Positive (semi-) definite matrices, Solving linear recurrences.

- Normed Linear Spaces: Normed spaces, Cauchy-Schwarz inequality and triangle inequality, Projection, Gram-Schmidt orthogonalization, Hermitian operators, The Spectral theorem.

- Matrix Computations: Floating point numbers and operations, Error Analysis, Solving systems of linear equations - Direct (Gaussian elimination, LU factorization) and Iterative methods (Jacobi method, Gauss-Seidel method), Solving least square problems - QR decomposition, Gram-Schmidt orthogonalization, Singular value-decomposition (SVD), Solving Eigenvalue problems - Tridiagonal QR iteration, Jacobi method.

- Some practical applications (if time permits).

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures.

Week	**Content**
Week 1	• Motivation:Why do we need to study LAMC? • A brief discussion of some practical application • Vector space, Subspace, Span, Linearly dependent & independent, Basis and Dimension
Week 2	• Linear maps/transformation, null space, range of linear maps and rank-nullity theorem, Injective, Surjective and Bijective Linear map, System of linear equations and relation with linear map, Matrix, Linear map represent by a matrix
Week 3	• Direct sum, Linear map represent by a matrix and introduction to the Matrix algebra, Invertibility and Isomorphic Vector Spaces
Week 4	• Products and Quotients of Vector Spaces, Linear functional, Dual Space and Dual Map
Week 5	• Matrix algebra, special matrices, row space and column space, Matrix rank and inverse • Assignment-1

Week	Content
Week 6	• Rank factorization and change of basis • Elementary operations and Echelon form • Assignment-2
Week 7	• Elementary operations and Reduced Echelon form • Normal form and Hermite canonical form
Week 8	• Linear equation • Assignment-3 • Course review for the mid-term examination
Week 9	• Sweep-out method for solving $Ax = b$
Week 10	• Determinants • Assignment-4
Week 11	• Inner product, norm and orthogonality, Cauchy–Schwarz inequality • Orthogonal complement, projection, orthogonal and unitary matrices
Week 12	• Eigenvalues and Eigenvectors • Matrix diagonalization
Week 13	• Singular value decomposition • Gaussian elimination, LU factorization and Cholesky factorization • Assignment-5
Week 14	• QR factorization, solving least square problem and Power method to find the largest eigenvalue and corresponding eigenvector
Week 15	• IEEE 754 floating point number and Error Analysis • Course review for the final examination

Machine Learning

Course Code: DA222

Course Description: This This course introduces the fundamental concepts and techniques of machine learning. Students will gain a solid understanding of various machine learning algorithms, their applications, and the process of building and evaluating machine learning models. Through lectures, discussions, hands-on exercises, and projects, students will develop the skills to apply machine learning to real-world problems.

Prerequisite(s): Student should have some knowledge in

- Mathematics: (some topic might need to discuss in brief)
  - Linear Algebra: *Vector space, Basis, Dimension, Matrix algebra (Addition, Multiplication, Trace, Inverse etc.), Eigen value and Eigen vectors, Positive definite matrices, Singular value decomposition etc.*
  - Multivariate Calculus: *Derivative, Partial derivative, Taylor series expansion, Chain rules etc.*
  - Basic Optimisation: *Convex set, Convex hull, Convex function, Gradient of a function, Hessian, Constrained and Unconstrained optimisation problem, Optimality condition*
  - Probability: *Definition, Random variables, Distribution function and their different variants, Conditional probability, Independence, Expectation, Variance, Moments, Entropy, Low of large numbers, Central limit theorem*
- Computer programming: *Any one from C/C++/Python (recommended for the class project and assignments)/MATLAB/Octave*
- Basic concept in Algorithms and Data Structure

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Credit Hours: 4

Text(s):

Machine Learning: a Probabilistic Perspective;
Kevin Patrick Murphy

Probabilistic Machine Learning: An Introduction;
Kevin Patrick Murphy

Probabilistic Machine Learning: An Introduction;
Kevin Patrick Murphy

Pattern Recognition and Machine Learning;
Christopher M. Bishop

Foundations of Machine Learning;
Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar

Understanding Machine Learning: From Theory to Algorithms;
Shai Shalev-Shwartz and Shai Ben-David

Statistical Learning with Sparsity: The Lasso and Generalizations;
Trevor Hastie, Robert Tibshirani and Martin Wainwright

airness and Machine Learning: Limitations and Opportunities;
Solon Barocas, Moritz Hardt and Arvind Narayanan

Deep Learning;
Ian Goodfellow, Yoshua Bengio, and Aaron Courville

Pattern Classification and Scene Analysis;
R. O. Duda, P. E. Hart and D. G. Stork

Pattern Recognition;
S. Theodoridis and K. Koutroumbas

Introduction to Statistical Pattern Recognition;
K. Fukunaga

A Probabilistic Theory of Pattern Recognition;
Luc Devroye, Laszlo Gyorfi, and Gabor Lugosi

Course Objective: Students will get to know

(1) the core concepts of machine learning, including supervised learning and unsupervised learning

(2) data preprocessing techniques to prepare data for machine learning models

(3) the appropriate machine learning algorithms for different types of problems (classification, regression, clustering, etc.)

(4) train and evaluate machine learning models through hands-on experience using Python and machine learning libraries
(5) eigenvalues and eigenvectors and their applications
(6) the performance of different machine learning models and identify potential biases or errors

Course outcomes: After successful completion of this course, students will be able to:
(1) explain the fundamental concepts of machine learning, including the different types of problems (supervised, unsupervised) and their applications
(2) apply machine learning algorithms such as linear regression, logistic regression, decision trees, and clustering algorithms to solve regression and classification problems
(3) identify the appropriate machine learning algorithms for various tasks, such as classification, regression, or clustering, based on the problem type and data characteristics
(4) implement machine learning algorithms using the Python programming language and popular libraries such as NumPy, pandas, and scikit-learn
(5) evaluate and interpret machine learning models using appropriate performance metrics such as accuracy, precision, recall, F1-score, and confusion matrices
(6) analyze and interpret machine learning results, identify patterns and insights in data, and make informed decisions based on model predictions and performance
(7) discuss the potential and limitations of machine learning models

Grade Distribution:

Assignments and Class Tests	35%
Midterm Exam	15%
Final Exam	50%

Course Outline (tentative) and Syllabus:

The tentative syllabus is as follows:

- Motivation: What is Machine Learning (ML) and why do we need to study ML?
- Data: Representation/Featurization, Normalization (after some discussion in classification/regression), Data partition (train, val, and test)

- Regression: Linear, Ridge, LASSO

- Classification: kNN, Bayes classifier, Linear discriminant analysis, Logistic regression, SVM, Decision tree, Random forest, Boosting, Ensemble methods

- Clustering: K-means, Hierarchical and agglomerative clustering/linkage clustering, Spectral graph clustering

- Dimensionality reduction and data visualization: PCA, Multidimensional scaling, Random projection, Isomap, t-SNE, UMAP, etc.

- Kernel methods: Definition, Reproducing Kernel Hilbert space, Kernel-SVM, Kernel-PCA, Kernel-Least square regression

- Low-rank matrix completion and compressive sensing

- ML and Society: Fairness, Explainability, and Environmental effects

- Learning theory: Approximation and estimation error, Empirical risk minimization, Convergence and consistency, Capacity measure of function classes, Shattering coefficient, VC dimension, Rademacher complexity, Occam's razor

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures. Reading assignments are from reference textbook will be mentioned in the suggested reading material in the *course url*.

Week	Content
Week 1	• Motivation: What is Machine Learning (ML) and why we need to study ML? • A brief discussion of some practical problems (research and industry) • Data: Representation/Featurization, Normalization (after some idea in classification/regression), Data partition (train, val and test)
Week 2	• Classification: kNN and Assignment-1 • Introduction Bayes classifier
Week 3	• Bayes classifier, Linear discriminant analysis • Assignment-2
Week 4	• Regression: Linear, Ridge, LASSO • Assignment-3 • Class project discussion
Week 5	• Linear classifier (perceptron) and Assignment-4 • Support Vector Machine (SVM)
Week 6	• Decision tree • Project proposal/Class test

Week	Content
Week 7	• Random forest, Boosting, Ensemble methods • Assignment-5
Week 8	• Clustering: K-means, Hierarchical and agglomerative clustering/linkage clustering, Spectral graph clustering • Assignment-6 • Project update • Course review for the mid-term examination
Week 9	• Dimensionality reduction and data visualization: PCA, Multidimensional scaling, Random projection, Issomap, t-SNE, UMAP etc. • Assignment-7
Week 10	• Kernel methods: Definition, Reproducing Kernel Hilbert space, kernel-SVM, kernel-PCA, kernel-Least square regression • Assignment-8 • Project update
Week 11	• Low rank matrix completion and compressive sensing
Week 12	• ML and Society: Fairness, Explainability and Environment effect • Project update
Week 13 - 15	• Learning theory: Approximation and estimation error, Empirical risk minimization, Convergence and consistency, Capacity measure of function classes, Shattering coefficient, VC dimension, Rademacher complexity, Occam's razor • Course review for the final examination

Multivariate Statistics

Course Code: DA310

Course Description: DA310 is going to provide an introduction to basic methods for analysis of multivariate data. Students will also learn to make practical use of the statistical computer package R.

Prerequisite(s): NA
Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Credit Hours: 4

Text(s):

Applied Multivariate Statistical Analysis ;
Richard A Johnson and Dean W. Wichern

An Introduction to Applied Multivariate Analysis with R;
Brian Everitt and Torsten Hothorn

Course Objective: Students will get to know
(1) the range of multivariate techniques available and their basic applications in real world,
(2) the link between multivariate techniques and corresponding univariate techniques,
(3) multivariate statistical analysis in R.

Course Outcome: The students will be able to
(1) summarize and interpret multivariate data,
(2) use multivariate techniques appropriately,
(3) undertake multivariate hypothesis tests, and draw appropriate conclusions,
(4) perform multivariate statistical analyses using R.

Grade Distribution:

Assignments and Class Tests	30%
Midterm Exam	20%
Final Exam	50%

Course Outline (tentative) and Syllabus:
The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures. Quizzes will be unannounced.

Week	Content
Week 1	• Introduction • Multivariate data representation
Week 2	• Mean Vector • Variance-covariance matrix • Sample mean vector and covariance matrix
Week 3	• Multivariate normal distribution • Properties of multivariate normal
Week 4	• Quiz • Maximum likelihood estimation • Sampling Distribution
Week 5	• Assessing normality • Transformation to normal
Week 6	• Inference about mean vector • Hotelling T^2
Week 7	• Multivariate confidence region • Different types of confidence interval
Week 8	• Inference about mean vector with missing observations • Problem Session • Review for Midterm exam
Week 9	• Comparisons of Several Multivariate Means – Paired Comparisons – A Repeated Measures Design for Comparing Treatments
Week 10	• Comparing Mean Vectors from Two Populations • One-Way MANOVA
Week 11	• Principle component analysis • Population principal component
Week 12	• Summarizing sample variation by PCA • Graphing principal component
Week 13	• Factor analysis • Orthogonal factor models
Week 14	• Factor estimation • Factor rotation • Quiz 2
Week 15	• Problem Session • Review for Final Exam

Mining of Massive Datasets

Course Code: DA207

Course Description: The course will discuss data mining and machine learning algorithms for analyzing very large amounts of data. The course will cover the fundamental data mining techniques like finding similar items, link analysis, frequent item sets, clustering, advertising on the web, recommendation systems etc. These techniques have huge applications in the diverse field of science, economics, agriculture etc. and are heavily used in industry for solving many real life problems.

Prerequisite(s): Students are expected to have the basic knowledge of algorithm analysis, linear algebra, probability and statistics.
Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Credit Hours: 4

Text(s):

Mining of Massive Datasets;
Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman

Probabilistic Machine Learning: An Introduction;
Kevin Patrick Murphy

Data Mining: Concepts and Techniques;
Jiawei Han, Micheline Kamber and Jian Pei

Introduction to Data Mining;
Tan, Steinbach and Kumar

Foundations of Data Science;
Avrim Blum, John Hopcroft, and Ravindran Kannan

Course Objective: Students will get to know

(1) To fully understand standard the data mining methods and techniques such as association rules, data clustering and classification
(2) Learn how the search engine crawls the web pages and rank them, advertising on the web work
(3) Lean how e-Commerce industries increase their productivity using frequent item sets mining

(4) Learn how plagiarisms are found by plagiarism softwares
(5) Lean how Amazon, Flipkart etc. recommend their products which are likely to be bought by the customers
(6) Learn new, advanced techniques for emerging applications (e.g. social network nalysis, stream data mining)

Course outcomes: After successful completion of this course, students will be able to:
(1) Students should be proficient in implementing and applying scalable algorithms and techniques for processing and analyzing massive datasets using frameworks such as MapReduce and Spark
(2) Students should be able to apply a variety of data mining algorithms, including classification, clustering, association rule mining, and anomaly detection, to extract insights from large datasets
(3) Students should be able to apply machine learning algorithms and techniques that are scalable and suitable for large datasets, such as online learning and parallelized algorithms
(4) Students should be able to analyze large-scale graphs and networks using graph mining techniques, including community detection, centrality analysis, and recommendation systems

Grade Distribution:

Assignments and Class Tests	20%
Midterm Exam	30%
Final Exam	50%

Course Outline (tentative) and Syllabus: The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures.

Week	Content
Week 1	• Introduction to Mining of Masive Datasets • Introduction to MapReduce architecture
Week 2	• MapReduce: Algorithms Using MapReduce • Finding similar Items: Applications of Set Similarity, Shingling of documents, Similarity-Preserving Summaries of Sets
Week 3	• Finding similar Items: Locality-Sensative Hashing(LSH), Distance Measures, Theory of Locality-Sensitive Functions, LSH Families for Other Distance Measures
Week 4	• Mining Data Streams: Stream Data Model, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream
Week 5	• Mining Data Streams: Estimating Moments, Counting Ones in a Window, Decaying Windows
Week 6	• Link Analysis: PageRank, Efficient Computation of PageRank,
Week 7	• Link Analysis: Topic-Sensitive PageRank, Link Spam, Hubs and Authorities
Week 8	• Frequent ItemSets: Market-Basket Model, A-Priori Algorithm, Compact Representation of Frequent Itemsets,
Week 9	• Frequent ItemSets: Alternative Methods for Generating Frequent Itemsets, FP-Growth Algorithm
Week 10	• Advertising on the Web: On-Line Advertising, On-Line Algorithms, The Matching Problem,
Week 11	• Advertising on the Web: The Adwords Problem, Adwords Implementation
Week 12	• Recommendation Systems: Model for Recommendation Systems, Content-Based Recommendations, Collaborative Filtering, Dimensionality Reduction
Week 13	• Mining Social Networks: Social Networks as Graphs, Clustering of Social-Network Graphs, Direct Discovery of Communities
Week 14	• Mining Social Networks: Partitioning of Graphs, Finding Overlapping Communities, Simrank, Counting Triangles, Neighborhood Properties of Graphs
Week 15	• Dimensionality Reduction: Principal-Component Analysis, UV-Decomposition. Singular-Value Decomposition, CUR decompostion

Natural Language Processing

Course Code: DA243

Course Description: TThis course presents an introduction to the field of computational models of written natural language, known as **Natural Language Processing** (NLP). We will explore both classical statistical and modern neural network-based language models. The course will cover theories, algorithms, and implementation (hands-on) of NLP models, aiming to highlight areas of ongoing research.

Prerequisite(s): Student should have some knowledge in

- Mathematics: *Linear Algebra, Multivariate Calculus, Basis Optimisation and Basic probability*
- Computer programming: *Any one from C/C++/Python (recommended for the class project and assignments)/MATLAB/Octave*
- Basic concept in Algorithms and Data Structure
- Introduction to Machine Learning

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Credit Hours: 4

Text(s):

Speech and Language Processing;
Dan Jurafsky and James H. Martin

Introduction to Natural Language Processing;
Jacob Eisenstein

Foundations of Statistical Natural Language Processing;
Chris Manning and Hinrich Schütze

Deep Learning;
Ian Goodfellow, Yoshua Bengio, and Aaron Courville

Course Objective: Students will get to know
(1) the core concepts of NLP, including language representation, tokenization,

and stemming/lemmatization
(2) techniques for text preprocessing to clean and prepare textual data for NLP tasks
(3) common NLP algorithms for tasks like sentiment analysis, named entity recognition, and machine translation
(4) advanced topics in NLP, including deep learning models for sequence processing
(5) use of different NLP toolkits (like NLTK, PyTorch etc.) to solve practical NLP tasks
(6) the performance of NLP models and identify potential biases or errors

Course outcomes: After successful completion of this course, students will be able to:
(1) explain textual data and extract representative features
(2) understand the basic concepts and basic algorithms of NLP
(3) understand the mathematical basis of various deep-learning based language models
(4) implement NLP algorithms and techniques in PyTorch for some of the major tasks in NLP
(5) decompose a real-world problem into subproblems in NLP and identify potential solutions
(6) discuss the potential and limitation of NLP (large language) models

Grade Distribution:

Assignments and Class Tests	35%
Midterm Exam	15%
Final Exam	50%

Course Outline (tentative) and Syllabus:

The tentative syllabus is as follows:

- What is Natural Language Processing (NLP) and why do we need to study NLP?
- Language Models (classical): *n-grams, smoothing, class-based, brown clustering* etc.
- Distributional Semantics: *distributional hypothesis, vector space models*, etc.
- Sequence Labelling ((classical)): *HMM, CRFs*, and applications of these models in *Part of Speech (POS)* tagging, *Named-entity recognition (NER)* etc.

- Language theory: *Grammar, Regular expression, Finite state acceptors (FSA), Context-free grammar (CFG), Probabilistic Context-free grammar (PCFG), Constituency (CKY) Parsing, Dependency parsing* etc.
- Application: *text classification*, *Part of Speech (POS)* tagging, *Named-entity recognition (NER). Coreference Resolution*, some NLP toolkits, etc.
- Neural language model:
  - *Neural Networks (NN), Activation functions, Backpropogation, Softmax, Cross-entropy, Gradient descent, Stochastic Gradient descent, Layer Normalization, Dropout*, etc.
  - Word representation (Vector): *Feed-forward NN, Word2Vec, GloVE,* etc.
  - Sequence modelling: *Recurrent neural network (RNN), Long short-term memory (LSTM), Self-Attention network*, etc.
  - Transformers: *Attention, BERT* and its different variants, Encoder-Decoder models
  - Large language model (LLM): *GPT* different variants, *pre-trained language model, transfer learning* and application

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures. Reading assignments are from reference textbook will be mentioned in the suggested reading material in the *course url*.

Week	Content
Week 1	• Motivation: What is Natural Language Procession (NLP) and why we need to study NLP? • A brief discussion of some practical problems (research and industry) • Text document classification (Bag-of-Words model) and Assignment-1
Week 2	• Language model: N-gram, Smoothing • Assignment-2
Week 3	• Language model: class-based and brown clustering
Week 4	• Word embeddings: vector semantics, Word2Vec • Assignment-3 • Class project discussion
Week 5	• Sequence labelling: Part-of-speech tagging and named entities recognition • HMM, CRF
Week 6	• Formal language theory • Context-free parsing and Assignment-4
Week 7	• Class project proposal • Probabilistic Context-free grammar (PCFG), Constituency (CKY) Parsing
Week 8	• Dependency parsing • Assignment-5 • Course review for the mid-term examination
Week 9 - 15	• Neural language model: – *Neural Networks (NN), Activation functions, Backpropogation, Softmax, Cross-entropy, Gradient descent, Stochastic Gradient descent, Layer Normalization, Dropout*, etc. – Word representation (Vector): *Feed-forward NN, Word2Vec, GloVE,* etc. – Sequence modelling: *Recurrent neural network (RNN), Long short-term memory (LSTM), Self-Attention network*, etc. – Transformers: *Attention, BERT* and its different variants, Encoder-Decoder models – Large language model (LLM): *GPT* different variants, *pre-trained language model, transfer learning* and application • Assignment-6 & 7 55 • Course review for the final examination

Optimization for Machine Learning

Course Code: CS211

Course Description: This course focuses on select topics from convex optimization theory, algorithms of unconstrained and constrained optimization, batch gradient methods etc. that are needed to build a solid understanding of, as well as implement several machine learning algorithns. The purpose of this course is to summarize and analyze classical and modern optimization methods from a machine learning perspective.

Prerequisite(s): Linear Algebra, Multivariate calculus

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Credit Hours: 4

Text(s):

Optimization for Machine Learning ;
Suvrit Sra, Sebastian Nowozin and Stephen Wright

Nonlinear Programming: Theory and Algorithms;
M. S. Bazaraa, Hanif D. Sherali, C. M. Shetty

Convex Optimization;
Stephen Boyd and Lieven Vandenberghe

Course Objective: Students will get to know

(1) Theory of convex functions, sets and convex optimization
(2) Geometric interpretation of feasibility and optimality conditions
(3) Conditions for existence and uniqueness of optimal solution
(4) Different unconstrained and constrained algorithms used in Machine Learning models

Course Outcome: The students will be able to

(1) How to formulate optimization problems
(2) How to write a computer program to solve a mathematical optimization problem
(3) How to perform sensitivity analysis on the optimal solution of a problem
(4) Ability to understand the interplay between optimization and machine

learning

Grade Distribution:

Assignments and Class Tests	20%
Midterm Exam	30%
Final Exam	50%

Course Outline (tentative) and Syllabus:

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures. Quizzes will be unannounced.

Table 1: **Topics**

Mathematical Preliminaries

- Norms of vectors, Sets and Functions in euclidean space
- Derivatives of multivariate functions, directional derivative, Gradient, Jacobian and Hessian

Theory of Convex Optimization

- Convex sets, convex functions and their characterizations
- Convexity preserving operations
- Composition of convex functions
- Generalized convexity
- Unconstrained Optimization: Necessary and Sufficient conditions
- Constrained Optimization: Necessary and Sufficient conditions for problems with equality and inequality constraints, KKT optimality conditions, Constraint Qualification, Lagrangian Duality and Saddle Point Optimality Criteria

Optimization in ML : – Analysis of ML model(s) from the point of Convex Optimization Theory

Optimization Algorithms used in Machine Learning

- First-order Algorithms: Steepest/Gradient Descent, Properties of GD
- (if time permits) Conjugate Gradient, Stochastic Gradient Descent

Java and Hadoop (JH)

Course Code:DA110 Credit:4

Course Description:

This course explores the essential components of Distributed Computing, including file systems and data processing. It covers Hadoop architecture in detail. As a prerequisite, the course introduces Java programming concepts with focus on IO and networking. It addresses standard MapReduce problems, such as Join, Top N pattern, Partitioner, Combiner, Sort, and Group Comparator, with limited application to machine learning and related fields. Throughout the course, there will be extensive implementation of these concepts, not only enhancing students' programming skills for solving real-world problems but also sharpening their analytical and critical thinking abilities.

Prerequisite(s): Basic Programming Skills in any language

Note: Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. Students, however, will be evaluated only on the basis of topics covered in the course.

Course Objectives:

- Introduce programming in Java
- Introduce Distributed Computing Fundamentals
- Provide hands-on experience in building elements of a distributed system framework
- Conduct a high-level source code walkthrough of the Hadoop framework

Course Outcomes:

- Develop understanding on applications of Distributed Computing in Practice
- Gain hands on practice and understanding on operating environment on UNIX system (prerequisite for distributed computing)
- Develop theoretical understanding of File System and basic OS concepts (prerequisite).
- Gain understanding and practice on Hadoop Distributed File System.
- Acquire a foundational understanding of the Java programming language, including syntax, I/O, networking, and object-oriented programming (OOP).

- Gain understanding on Hadoop (MapReduce pipeline) fundamentals with hands on problem solving (approx 6-8 problems covering groupby/topN, partitioner, sort and group comparator, ML examples)

Approximate weights of different components in evaluation (this is modified based on interest and orientation of the students):

Assignments/Tests	20%
Midterm Exam	30%
Final Exam	50%

Course Policies:

General

1. Computing devices are not to be used during any exams unless instructed to do so.
2. Quizzes and exams are open books and open notes.
3. Practice sessions are planned for the second half of the course.
4. ***No makeup quizzes or exams will be given.***

Attendance and Absences

1. Attendance is expected and will be taken in each class. Students are not supposed to miss class without prior notice/permission.
2. Students are responsible for all missed work, regardless of the reason for absence. It is also the absentee's responsibility to get all missing notes or materials.

Textbooks(s):

1. Java: The Complete Reference
   Herbert Schildt
2. Hadoop The Definitive Guide 4^{th} edition
   Tom White
3. Data Algorithms - Recipes for Scaling Up with Hadoop and Spark, 1^{st} Edition
   Mahmoud Parsian

Course Outline (tentative) and Syllabus:

The weekly coverage might change as it depends on the progress of the class. Each week assumes 4 hour lectures and 2 hour practicals (as and when required). Quizzes will be unannounced, so students should maintain close to 100% attendance.

Week	Content
Week 1	• Objective of the course • Java Introduction - installation and Hello World program execution
Week 2	• Introduction to Distributed Computing and MapReduce
Week 3	• Java basic examples and Object Oriented Programming • Java Reflection
Week 4	• File IO and Networking part 1 • File System Introduction
Week 5	• File IO and Networking part 2 • HDFS
Week 6	• Java Generics • Java Exception Handling • Hadoop Resource Management and Configuration
Week 7	• Java Threads and ExecutorService • Hadoop Installation Part 1
Week 8	• Hadoop Installation Part 2 • Review of Java IO and Networking
Week 9	• Java Manifest file, JAR, maven, Serialization • Hands on HDFS and MapReduce
Week 10	• Combiner, Partitioner • TopN Pattern, Join part 1
Week 11	• Custom Reader and Writer • Join part 2 (DistributedCache)
Week 12	• Practice Sessions
Week 13	• Application of Hadoop - Common Friends, Market Basket Analysis • Application of Hadoop - KNN / KMeans
Week 14	• Hadoop Source Code Walkthrough • Introduction to related applications (Hive, Kafka)
Week 15	• Sort and Group Comparator • Comparison and Contrast with Other Architectures

Enabling Technologies for Big Data Computing (ETBDC)

Course Code: DA230 Credit:4

Course Description:

The course explores essential components of Distributed Computing, focusing on both Stream and Batch processing, along with Graph Databases. As a prerequisite, students are introduced to Java programming with an emphasis on I/O operations and networking. The curriculum covers standard MapReduce problems such as Join, Top N pattern, Partitioner, Combiner, Sort, and Group Comparator, with limited applications in machine learning and related areas. Stream processing is taught using Apache Spark, providing hands-on experience in real-time data handling. Extensive implementation of these concepts throughout the course will not only enhance students' programming skills for solving practical, real-world problems but also sharpen their analytical and critical thinking abilities. Toward the end of the course, students are introduced to NEO4J and CypherQL, focusing on graph algorithms, with a brief introduction to Graph Neural Networks.

Prerequisite(s): Basic Programming Skills in any language

Note: Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. Students, however, will be evaluated only on the basis of topics covered in the course.

Course Objectives:

- Introduce programming in Java
- Introduce Distributed Computing Fundamentals - batch and stream
- Provide a comparison and contrast with other Distributed Computing Architectures
- Introduce Graph Database (Neo4J) and Graph algorithms with Graph Neural Network

Course Outcomes:

- Develop understanding on operating environment on Ubuntu / Debian (prerequisite for distributed computing)
- Develop understanding on File System on Linux (prerequisite for distributed computing)

- Develop understanding on principles of general Distributed File Systems
- Develop understanding on hands on Hadoop Distributed File System
- Develop understanding on basic Java programming language (e.g. fundamentals of syntax, i/o, networking etc.) with OOP (prerequisite for distributed computing because most of the distributed computing systems are wriiten in java with OOP with a layer of python - exception Ray, only ML/DL solutions)
- Develop understanding on specific Java topics (Generics, Optionals, Reflection, Serialization etc)
- Develop understanding on Distributed batch and Stream Processing
- Develop understanding on Hadoop (MapReduce pipeline) fundamentals with hands on problem solving (approx 6-8 problems covering groupby/topN, partitioner, sort and group comparator, ML examples)
- Develop understanding on real time distributed stream processing
- Develop understanding on Spark with hands on batch processing
- Develop understanding on how to make comparison/contrast with multiple Distributed computing frameworks
- Develop understanding on stream processing with spark / pyspark with hands on examples on variations on window parameters
- Develop understanding of Complex Event Processing (CEP) with reference to streaming
- Develop understanding on database systems (hands on) with graph database fundamentals
- Develop understanding on Neo4J architecture with CypherQL
- Develop understanding on graph algorithms supported by Graph databse systems (hands on with Neo4J) - examples: PageRank, Centrality, Community Detection
- Develop understanding on application of Neo4J on recommendation systems, fraud detection system and primitive language models
- Develop understanding on Neo4J LLM interactions (theory)

Approximate weights of different components in evaluation (this is modified based on interest and orientation of the students):

Assignments	20%
Midterm Exam	30%
Final Exam	50%

Course Policies:

General

1. Computing devices are not to be used during any exams unless instructed to do so.
2. Quizzes and exams are open books and open notes.
3. Practice sessions are planned for the second half of the course.
4. ***No makeup quizzes or exams will be given.***

Attendance and Absences

1. Attendance is expected and will be taken in each class. Students are not supposed to miss class without prior notice/permission.
2. Students are responsible for all missed work, regardless of the reason for absence. It is also the absentee's responsibility to get all missing notes or materials.

Textbooks(s):

1. Java: The Complete Reference
   Herbert Schildt
2. Hadoop The Definitive Guide 4^{th} edition
   Tom White
3. Data Algorithms - Recipes for Scaling Up with Hadoop and Spark, 1^{st} Edition
   Mahmoud Parsian
4. Spark in Action, 2^{nd} Edition
   Jean-Georges Perrin
5. Neo4j in Action
   Jonas Partner, Aleksa Vukotic, and Nicki Watt
6. Graph-Powered Machine Learning
   Alessandro Negro

Course Outline (tentative) and Syllabus:

The weekly coverage might change as it depends on the progress of the class. Each week assumes 4 hour lectures and 2 hour practicals (as and when required). Quizzes will be unannounced, so students should maintain close to 100% attendance.

Week	Content
Week 1	• Objective of the course, Java Introduction - installation and Hello World program execution • Introduction to Distributed Computing and MapReduce
Week 2	• Java basic examples and Object Oriented Programming with Reflection • Java File IO and Networking part 1
Week 3	• File System Introduction and HDFS • Java Generics, Exception Handling, Threads
Week 4	• Hadoop Resource Management and Configuration • Hadoop Installation • Java Manifest file, JAR, maven, Serialization
Week 5	• Hands on HDFS and MapReduce • Combiner, Partitioner
Week 6	• TopN Pattern, Join • Custom Reader and Writer
Week 7	• Practice Sessions • Application of Hadoop - Common Friends, Market Basket Analysis, KNN / KMeans
Week 8	• Sort and Group Comparator • Hadoop: Comparison and Contrast with Other Batch Architectures • Motivation behind Stream Processing (Spark and Flink)
Week 9	• Batch Processing in Spark (RDD, DataFrame, Dataset) • Transformation, Action, User Defined Function
Week 10	• Spark SQL • Spark Graphframe
Week 11	• Spark Stream Processing Part 1 - Handling Windows • Spark Stream Processing Part 2 - Comparison with Flink (CEP)
Week 12	• Practice Sessions on Batch and Stream Processing in Spark
Week 13	• Graph Database Systems • Neo4J Architecture, tools and python interface (OGM)
Week 14	• Graph Algorithms (e.g. Community Detection, Centrality) • Practice Session on Neo4J and Spark GraphFrame
Week 15	• Applications: Recommendation Systems, Fraud Detection, Language Modelling • Graph Neural Network Overview

Probability and Stochastic Process

Course Code: DA104

Course Description: This course aims at providing a comprehensive review of probabilistic tools, techniques and basics of stochastic process. We shall focus more on applications of the techniques in problem solving.

Prerequisite(s): NA
Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Credit Hours: 4

Text(s):

A First Course in Probability ;
Shelden M. Ross

Introduction to Stochastic Process;
Paul G. Hoel, Sydney C. Port and Charles J. Stone

Mathematical Statistics and Data Analysis;
John A. Rice

Probability - Random Variables and Stochastic Processes;
Athanasios Papoulis, S. Pillai

Course Objective: Students will get to know

(1) different discrete and continuous probability distributions

(2) various standard statistics from mass, distribution and density functions from numerous univariate as well as bivariate distributions,

(3) different random processes to formulate and solve practical engineering problems..

Course Outcome: The students will be able to

(1) identify and interpret the key parameters which lies behind the random nature of the problems from various fields,

(2) convert a real-world problem into a precise mathematical probabilistic problem,

(3) use different statistical principles and the properties of random variables to solve various probabilistic problems.

Grade Distribution:

Assignments and Class Tests	20%
Midterm Exam	30%
Final Exam	50%

Course Outline (tentative) and Syllabus:

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures. Quizzes will be unannounced.

Week	Content
Week 1	• Basic Probability • Introduction to sample spaces • Classical definition of probability
Week 2	• Countable probability space • Counting methods using combinatorial tools • Conditional probability • Independence of events
Week 3	• Discrete Random Variables • Probability mass function • Expectation
Week 4	• Discrete Distribution • Binomial • Poisson • Geometric
Week 5	• Continuous random variables • Probability density function • Cumulative distribution function • Expectation
Week 6	• Continuous distribution • Uniform • Exponential • Normal
Week 7	• Joint distribution • Joint PMF for discrete random variables • Joint density and CDF
Week 8	• Independence of random variables • Conditional distribution and expectation • Covariance and correlation • Review for Midterm Exam

Week	Content
Week 9	• Central limit theorem • Characteristic function • Central limit theorem for finite variance
Week 10	• Weak law of large numbers for finite variance • Random walks
Week 11	• Markov chains • Definition and examples
Week 12	• Decomposition of states into communicating classes • Recurrence and transience as class properties
Week 13	• Invariant distributions for finite irreducible chains • Expected return time
Week 14	• Poisson processes • Definition and basic properties
Week 15	• Conditional distribution of arrival times given the number of arrivals • Non-homogeneous Poisson processes • Review for Final Exam

Reinforcement Learning

Course Code: DA344

Course Description: Reinforcement Learning (RL) is a branch of machine learning that deals with sequential decision-making. RL, however is different from other machine learning paradigms, since there is no supervisor for training. Instead, there is a trial-and-error learning process, that involves reward, feedback from environment, action, and time of action. The agent interacts with a dynamic, stochastic, and incompletely known environment, with the goal of finding an action-selection strategy, or policy, to optimize some measure of its long-term performance. RL has a wide range of applications in many problem situations, such as robotics, healthcare, smart grids, finance, self-driving cars etc., where explicit instructive signals are not available.The goal of the course is to introduce the mathematical foundations of reinforcement learning drawing from Operations Research, and do hands-on project on some small scale recent problems.

Prerequisite(s): Optimization Techniques, Matrix Algebra, Python

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Credit Hours: 4

Text(s):

Reinforcement Learning: An Introduction;
Richard S. Sutton and Andrew G. Barto, 2nd Edition, MIT Press, Cambridge, MA, 2018

Reinforcement Learning: State-of-the-Art;
Marco Wiering and Martijn van Otterlo, Eds. Springer-Verlag Berlin Heidelberg 2012

Artificial Intelligence: A Modern Approach;
Stuart J. Russell and Peter Norvig. Fourth edition, 2020

Deep Reinforcement Learning: Fundamentals, Research and Applications;
Hao Dong, Zihan Ding, Shanghang Zhang Ed.s © Springer Nature Singapore Pte Ltd. 2020

Deep Reinforcement Learning: Frontiers of Artificial Intelligence;

Course Objective: Students will get to know

(1) Reinforcement Learning (RL) is a general purpose formalism for automated decision-making where an agent explicitly takes actions and interacts with the environment. Understanding the importance and challenges of learning agents that make decisions is of vital importance in designing interactive AI agents. This course focuses on various important aspects of RL, viz., formulating problems as Markov Decision Processes, understanding the basic exploration methods and the exploration/exploitation tradeoff, understanding value functions and how to implement dynamic programming as an efficient solution approach, identifying the impact of choices on performance and validating the expected behaviour of algorithms

(2) Understanding of how RL relates to and fits under the broader umbrella of machine learning, deep learning, supervised and unsupervised learning.

(3) Understanding of the space of RL algorithms (Temporal- Difference learning, Monte Carlo, Q-learning, Policy Gradient etc.).

(4) Understanding of how to implement reinforcement learning in RL platforms such as, **TensorFlow/Open AI Gym, PyTorch** etc.

Grade Distribution:

Assignments and Class Tests	30%
Midterm Exam	20%
Final Exam	30%
Final Project	20%

Course Outline (tentative) and Syllabus:

The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments. Each week assumes 4 hour lectures. Quizzes will be unannounced.

Course Project :

There will be a course project, aimed at implementing reinforcement learning theory and algorithms to practical applications. It will also give students opportunity to learn how to deploy and train reinforcement learning on AI frameworks (e.g., PyTorch, OpenAI Gym, Tensorflow). The topic would be finalised based upon detailed discussion with the Instructor. An intermediate presentation of the project topic has to be done after midterm exam. Students can possibly extend and expand on the course project later on and give it a more concrete shape leading to term project and/or masters thesis.

Table 2: **Topics (for 16 weeks)**

A. Preliminaries (1 week)

- The Reinforcement Learning (RL) framework, RL vis-a-vis supervised/unsupervised ML
- Interesting applications of RL

B. Mathematical and Algorithmic Foundations of RL (4 weeks) – *The following topics will be covered in detail*

- *Sequential decision making*: Dynamic Programming Problem, Bellman Optimality Equations
- *Markov Decision Process* (MDP): State, stage, action, total expected reward criteria, average reward criteria. Optimal policies and optimal value functions. Value iteration and Policy Iteration. Optimal solution vs. approximate solution of Bellman optimality equation.
- *Partially Observable Markov Decision Process* (POMDP): The observation model, Monahan's enumeration algorithm, pruning via Linear Programming, the Witness algorithm, Parsimonious representation of value functions, near-optimal solutions for POMDPs

C. Methods of RL (8-10 weeks) – *The following topics will be covered in detail*

- *Model based methods*: Value iteration, Policy Iteration, Conditions on convergence Exploration/Exploitation Tradeoff (the k-armed bandit problem)
- *Model free methods*: Watkins' Q-learning,Sutton's Dyna architecture, Monte Carlo and Gaussian Process Temporal Difference Learning
- *Approximate solution methods*: Policy gradient, Monte Carlo policy gradient, Actor-critic algorithm, Performance evaluation of policy gradient methods

D. Frontiers in RL (1 week)- *If time permits, an overview will be given on the following topics:*

- Hierarchical Reinforcement Learning
- Game Theory and Multi-agent Reinforcement Learning
- Decentralised POMDP