UNSW Course Outline

ZEIT4151 Machine Learning - 2024

Published on the 27 Jun 2024

General Course Information

Course Code : ZEIT4151
Year : 2024
Term : Semester 2
Teaching Period : Z2
Is a multi-term course? : No
Faculty : UNSW Canberra
Academic Unit : School of Systems and Computing
Delivery Mode : In Person
Delivery Format : Standard
Delivery Location : UNSW Canberra at ADFA
Campus : UNSW Canberra
Study Level : Undergraduate
Units of Credit : 6

Useful Links

Handbook Class Timetable

Course Details & Outcomes

Course Description

ZEIT4151 (Machine Learning) is a 6 Units of Credits (UoC) course. This core course exposes engineering, computer science, and cyber security students to the various aspects of machine learning. The techniques the students learn in this course do apply to a wide spectrum of machine learning problems as well as practical data analysis applications.

Course Aims

The aim of this course is to expose students to different types of learning that an Artificial Intelligence agent could use and provide them with a hands-on experience with Machine Learning algorithms, including the analysis, design, implementation, and validation of Machine Learning algorithms.

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Map real-life problems into the appropriate mathematical model which machines can optimise and learn from.
CLO2 : Interpret and explain how the data affect the model learning experience and outcomes.
CLO3 : Judge the efficacy of machine learning algorithms and models to address the distinct characteristics of the problems and the associated data.
CLO4 : Develop systems that use machine learning to make intelligent decisions and evaluate their performance.

Course Learning Outcomes	Assessment Item
CLO1 : Map real-life problems into the appropriate mathematical model which machines can optimise and learn from.	• Feature Extraction • ML with Images • Final Assessment
CLO2 : Interpret and explain how the data affect the model learning experience and outcomes.	• Feature Extraction • ML with Images • Final Assessment
CLO3 : Judge the efficacy of machine learning algorithms and models to address the distinct characteristics of the problems and the associated data.	• Model Validation • ML with Images • Final Assessment
CLO4 : Develop systems that use machine learning to make intelligent decisions and evaluate their performance.	• ML with Images • Final Assessment

Learning and Teaching Technologies

Moodle - Learning Management System | Echo 360 | Microsoft Teams

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Model Validation Assessment Format: Individual Short Extension: Yes (5 days)	10%	Start Date: 18/07/2024 12:00 AM Due Date: 26/07/2024 12:00 AM
Feature Extraction Assessment Format: Individual Short Extension: Yes (5 days)	25%	Start Date: 09/08/2024 12:00 AM Due Date: 23/08/2024 12:00 AM
ML with Images Assessment Format: Individual Short Extension: Yes (5 days)	25%	Start Date: 09/09/2024 12:00 AM Due Date: 23/09/2024 12:00 AM
Final Assessment Assessment Format: Individual Short Extension: Yes (5 days)	40%	Start Date: 23/09/2024 12:00 AM Due Date: 01/11/2024 11:55 PM

Assessment Details

Model Validation

Assessment Overview

Two coding tasks covering model training and validation. Delivery is expected to be a Jupyter notebook (code and explanation).

Course Learning Outcomes

- CLO3 : Judge the efficacy of machine learning algorithms and models to address the distinct characteristics of the problems and the associated data.

Detailed Assessment Description

Submission requirements Eachstudent is expected to submit one python file. Third party Python modules (e.g. pandas, numpy, scipy, etc) are not allowed. Only pure python constructs (e.g. lists, dictionaries, tuples, classes, functions, etc) are allowed. The submission python file must include a description on how to run the code and a sample of test cases. Submissions will be accepted via Moodle. The submission file name must be the zID of the student.

Feedback Details: Video feedback will be provided on the returned work 2 weeks after the submission, together with the marking rubric; general feedback will be provided in class; you will receive the feedback in the form of a mark and the answer key.

Aligned CLOs:

- CLO3: Judge the efficacy of machine learning algorithms and models to address the distinct characteristics of the problems and the associated data.

Policy for Using of Generative Artificial Intelligence (AI) – such as ChatGPT – in this assignment:

SIMPLE EDITING ASSISTANCE is applied

For this assessment task, you may use standard editing and referencing software, but not generative AI. You are permitted to use the full capabilities of the standard software to answer the question (Microsoft Office suite and Grammarly are allowed).

If the use of generative AI such as ChatGPT is detected, it will be regarded as serious academic misconduct and subject to the standard penalties, which may include 00FL, suspension and exclusion.

Rationale

Machine learning and data science projects are usually affected with observer bias. It is not uncommon that early-career data scientists tend to jump to conclusions and mis-interpret the results. This, in return, can lead the decision maker to adopt a policy which can lead to suboptimal results. While this is a problem across the entire discipline, it is especially apparent with classification problems due to the differences between the loss function a machine learning algorithm is trying to minimise and what the loss value means to the human observer. As discussed in the classification lecture, the machine can optimise a mean squared error of the logistic function and thus learn how to classification data points. However, the reported error is not an intuitive indication on how accurate and precise the model is.

Task description

Code

In this assessment, you are asked to write a python code which calculates the following classification metrics.

- Accuracy
- Precision
- Sensitivity
- Specificity
- F1-score

The python code accepts two arrays of labels as input and returns 5 values for the metrics above. The first array, the ground truth , is the true labels provided by a subject matter expert. The second array is the predicted labels predicted by a classification machine learning model. The nature of the ML model is irrelevant to the metric. The idea here is that you should not care about the model itself but rather assess the results objectively. The output should be an iterable of 5 values (e.g., a tuple or a list) or a dictionary with keys “accuracy, precision, sensitivity, specificity, f1-score” and their corresponding calculated values. A sample of the inputs is described below.

Tests

Students must include test cases to demonstrate how their code handles common-cases and corner-cases. Corner cases can include but not limited to being a set of zeros or ones (a poor model which assigns all data point to only one class).

Submission details

Submission is via Moodle and constitutes of one python code file (.py) which includes the python function, comments on how to use it, and a sample of test cases.

Resources

Students are allowed to explore internet sources including large language model and generative AI (e.g., ChatGPT and Co-Pilots). If a generative AI tool was used, the student must declare this in the submission and detail any bugs produced by the tool.

Submission notes

One Jupyter notebook and/or python file.

Assignment submission Turnitin type

Not Applicable

Feature Extraction

Assessment Overview

Students will collaborate in groups of four students. Each group will attempt working on four realistic datasets (one student per dataset). Delivery is expected to be a Jupyter notebook per dataset (code and explanation).

Course Learning Outcomes

- CLO1 : Map real-life problems into the appropriate mathematical model which machines

can optimise and learn from.
- CLO2 : Interpret and explain how the data affect the model learning experience and outcomes.

Detailed Assessment Description

Submission requirements Upload via Moodle

Feedback Details: Video feedback will be provided within 2 weeks after the submission due date.

Aligned CLOs:

- CLO1: Map real-life problems into the appropriate mathematical model which machines can optimise and learn from.
- CLO2: Interpret and explain how the data affect the model learning experience and outcomes.

Policy for Using of Generative Artificial Intelligence (AI) – such as ChatGPT – in this assignment:

SIMPLE EDITING ASSISTANCE is applied

For this assessment task, you may use standard editing and referencing software, but not generative AI. You are permitted to use the full capabilities of the standard software to answer the question (Microsoft Office suite and Grammarly are allowed).

If the use of generative AI such as ChatGPT is detected, it will be regarded as serious academic misconduct and subject to the standard penalties, which may include 00FL, suspension and exclusion.

Assessment Description

Real data are usually messy, unbalanced, and incomplete. In this assessment, we will experience working with real life data.

The objectives of this assessment are to

- become familiar with real datasets;
- challenge yourself to make hard decisions about your data; and
- become familiar with data from other application domains (it's just numbers and labels).

- **Task description**

Conduct feature selection and reduction for the following 4 datasets;

- Intrusion Detection
    - Goal: Classify network traffic from logged data
    - Features: Network traffic logs
    - Download: https://www.kaggle.com/datasets/subhajournal/sdn-intrusion-detection

- Explosive Detection
    - Goal: Detect if the material of an object of interest is explosive or not
    - Features: Material composition
    - Download: https://www.kaggle.com/datasets/saurabhshahane/explosive-detection

- Bankruptcy Detection
    - Goal: Detect if a company is close to bankruptcy
    - Features: Financial stats of the company
    - Download: https://www.kaggle.com/code/marto24/bankruptcy-detection/data?select=data.csv

- Bad Loan Detection
    - Goal: Identify if a bank loan is good or bad
    - Features: Financial wellbeing stats of the applicant
    - Download: https://www.kaggle.com/datasets/ikpeleambrose/irish-loan-data

Submission details

Each group is expected to submit a folder that has 4 mark-down formatted Jupyter notebooks (one for each dataset) detailing the following

1. Problem description
2. Thought process
3. Types of data fields
4. Features to drop
    - Explain the rationale behind the decision
    - Include the code
5. Features ranked by importance derived from decision trees
    - Include the code
    - Report results and feature scores
    - Explain the insight from the problem
6. A reasonable dimensionality reduction method
    - Describe number of components preserved
    - Report accuracy of reconstructing the data from selected components
7. Justify your decision using cross-validation with proper metrics
    - Demonstrate how your decisions improved the performance of the model (e.g., report

scores before and after feature dropping)

Resources and Hints

- The datasets used for this assignment are relatively large. This will require that you explore different methods to select the features that count. You may also need to explore different visualisation methods.
- Explore what other data scientists did to work with these datasets.
- Using Jupyter Notebooks will make it easier to export a quick report with code.

Assignment submission Turnitin type

Not Applicable

ML with Images

Assessment Overview

Individual students will develop machine learning models on a realistic image dataset. Delivery is expected to be a Jupyter notebook (code and explanation).

Course Learning Outcomes

- CLO1 : Map real-life problems into the appropriate mathematical model which machines can optimise and learn from.
- CLO2 : Interpret and explain how the data affect the model learning experience and outcomes.
- CLO3 : Judge the efficacy of machine learning algorithms and models to address the distinct characteristics of the problems and the associated data.
- CLO4 : Develop systems that use machine learning to make intelligent decisions and evaluate their performance.

Detailed Assessment Description

Project Narrative

This project aims at immersing the students in a practical experience with using machine learning to learn from different data modalities. For the selected modality modalities, the student would need to map the real-life problem definition to the appropriate ML problem type, demonstrate a good understanding of the characteristics of the dataset, and accordingly suggest ML algorithms that would suit each problem/dataset combination. Then, the student would practice developing a system that can extract meaningful information from the data that help address the problem definition. Presenting the process and findings in both the report and the presentation would help students practice ML engineers’ experience of communicating their findings to both technical and non-technical stakeholders.

The narrative of this project is about building a human-machine interface (HMI) for Amyotrophic Lateral Sclerosis (ALS) patients. ALS is a neurodegenerative disease that affects the spine and stops communication between the brain and the skeletal muscles. This renders the patient immobile except for the ocular muscles. Perhaps the most famous patient diagnosed with this rare disease was the late Prof Stephen Hawking.

Imagine developing three ML modules for an HMI system that allows communication via eye movements and brain waves. The system relies on eye-gaze tracking to select objects and letters on the screen (modules 1A and 1B). This will obviously allow very low-resolution mouse movements, which will be good enough for large objects but not for text or sub-objects. This is where we will need to augment this solution with EEG signals to have finer control of smaller objects/regions like letters (modules 2A and 2B). Finally, typing one letter at a time is very slow, and thus we will need a predictive text model which can suggest the next word given the context of the previous text (modules 3A and 3B).

Assessment Description: Eye-Gaze Tracking

Develop two models for eye gaze tracking. The first model (Module 1A) is a classifier for detecting where the gaze point lands on the screen (split in coarse grid of blocks). The second model (Module 1B) is a regressor for finer localising within the block detected by the first model (Module 1A).

- **Module 1A: Eye-Gaze Tracking Module**
  - Develop an ML module that estimates the azimuth and altitude angles of a human eye from cropped eye images.
  - Inputs (X's): Cropped eye-images (left and right eyes)
  - Outputs (Y's): azimuth and altitude angles (in radians)
  - Performance:
    - Error (RMSE): 0.05 (in radians)
    - R2: Minimum 65%, typical 75~85%, ideal 95%
    - FPS: Minimum 15 FPS, typical 25~35 FPS, ideal ~60FPS
  - Dataset: MPII Eye-Gaze Dataset (https://bit.ly/3CJlvPJ).
- **Module 1B: Eye-Gaze Tracking Module**
  - Develop an ML module that determines which square on the screen the eye gaze is looking at from cropped eye images.
  - Inputs (X's): Cropped eye-images (left and right eyes)

- Outputs (Y’s): Square ID
- Performance:
  - Error (F1-Score): Minimum 70%, typical 75~85%, ideal 95%
  - FPS: Minimum 15 FPS, typical 25~35 FPS, ideal ~60FPS
  - No. of squares: Minimum 9 (3wx3h), typical 12 (4wx3h), ideal 24 (6wx4h)
- Dataset: MPII Eye-Gaze Dataset (https://bit.ly/3CJlvPJ).

Submission requirements Uploadvia Moodle

Feedback Details: Video feedback will be provided within 2 weeks after the submission due date.

Aligned CLOs:

- CLO1: Map real-life problems into the appropriate mathematical model which machines can optimise and learn from.
- CLO2: Interpret and explain how the data affect the model learning experience and outcomes.
- CLO3: Judge the efficacy of machine learning algorithms and models to address the distinct characteristics of the problems and the associated data.
- CLO4: Develop systems that use machine learning to make intelligent decisions and evaluate their performance.

Policy for Using of Generative Artificial Intelligence (AI) – such as ChatGPT – in this assignment:

SIMPLE EDITING ASSISTANCE is applied

For this assessment task, you may use standard editing and referencing software, but not generative AI. You are permitted to use the full capabilities of the standard software to answer the question (Microsoft Office suite and Grammarly are allowed).

If the use of generative AI such as ChatGPT is detected, it will be regarded as serious academic misconduct and subject to the standard penalties, which may include 00FL, suspension and exclusion.

Task description

Explain the thought process during the development of the ML models. Please emphasise the following aspects.

1. **Dataset**

    1. What are the characteristics of this dataset in terms of?
    2. Does this dataset require pre-processing? If so, explain why these steps are needed and attach the snippet of the code for it.
    3. Do you suggest dimensionality reduction for this dataset? And why?
    4. Highlighting challenges with the dataset, certain samples or certain features.
    5. Consult the documentation of the dataset and the data collection procedure

2. **ML Model**

    1. How are you going to split the data?
    2. Select the 2 relevant algorithms/model that you would like to apply to this problem, and explain why you selected each of them?
    3. Reflect on your selection and rank them according to the expected performance for this type of problem. Justify your ranking.

3. **Results**

    1. What are the default parameters for this model?
    2. What are the results achieved using these parameters?
    3. Test a different set of parameters.
    4. Compare the performance of the different parameters.
    5. Describe your method to reach the best (or a good) set of parameters for this model.
    6. Write a summary of your final parameter selection for each of them.
    7. Choose the best tabular and graphical representations that can communicate and compare the performance of these models and their parameter tuning according to the identified metrics.

4. **Code**

    1. The code to be written in Python programming language
    2. The delivery of the code to be 1 separate notebook for the selected module.
    3. Code Readability

        1. Inline comments matching the verbal description
        2. Meaningful variable names
        3. Docstrings

    4. All the results should be reproducible.
    5. All the figures are to be created automatically in a folder named Figures, and each figure

should have a meaningful filename that starts with the figure number in the report and is followed by a descriptive name.

6. The scores and or numerical values that were reported in the table should be reported in either .txt or .csv files, where the results should always be preceded by a text line that explains what results would follow.
7. The performance of the model, including the tasks performance metrics and the model size, speed, and memory requirements to be exported automatically in a text file.

Submission details

The students are expected to submit a mark-down formatted Jupyter notebooks detailing the code and description of the thought process and the adopted methods/models/algorithms structured into the following high-level sections.

1. Dataset
2. ML Models
3. Results
4. Final Recommendations (3 sentences)

The notebook should be self-contained with all code, description and produced figures, graphs.

Resources and Hints

- The datasets used for this assignment are relatively large. This may require that you explore different methods to select the features that count. You may also need to explore different visualisation methods.
- Explore what other data scientists did to work with these datasets.
- Using Jupyter Notebooks will make it easier to export a quick report with code.

Submission notes

6-8 pages approximately including text, code, figures, tables, and references. Excludes debugging output.

Assignment submission Turnitin type

Not Applicable

Final Assessment

Assessment Overview

Programming task, video presentation, and report

Course Learning Outcomes

- CLO1 : Map real-life problems into the appropriate mathematical model which machines can optimise and learn from.
- CLO2 : Interpret and explain how the data affect the model learning experience and outcomes.
- CLO3 : Judge the efficacy of machine learning algorithms and models to address the distinct characteristics of the problems and the associated data.
- CLO4 : Develop systems that use machine learning to make intelligent decisions and evaluate their performance.

Detailed Assessment Description

Project Narrative

This project aims at immersing the students in a practical experience with using machine learning to learn from different data modalities. For the selected modality modalities, the student would need to map the real-life problem definition to the appropriate ML problem type, demonstrate a good understanding of the characteristics of the dataset, and accordingly suggest ML algorithms that would suit each problem/dataset combination. Then, the student would practice developing a system that can extract meaningful information from the data that help address the problem definition. Presenting the process and findings in both the report and the presentation would help students practice ML engineers' experience of communicating their findings to both technical and non-technical stakeholders.

The narrative of this project is about building a human-machine interface (HMI) for Amyotrophic Lateral Sclerosis (ALS) patients. ALS is a neurodegenerative disease that affects the spine and stops communication between the brain and the skeletal muscles. This renders the patient immobile except for the ocular muscles. Perhaps the most famous patient diagnosed with this rare disease was the late Prof Stephen Hawking.

Imagine developing three ML modules for an HMI system that allows communication via eye movements and brain waves. The system relies on eye-gaze tracking to select objects and letters on the screen (modules 1A and 1B). This will obviously allow very low-resolution mouse movements, which will be good enough for large objects but not for text or sub-objects. This is where we will need to augment this solution with EEG signals to have finer control of smaller objects/regions like letters (modules 2A and 2B). Finally, typing one letter at a time is very slow, and thus we will need a predictive text model which can suggest the next word given the context of the previous text (modules 3A and 3B).

Assessment Description: Implement module 2 (2A or 2B) and module 3 (3A or 3B), Video

Presnetation and Report

Module 2: EEG SSVEP Recognition [IMPLEMENT ONLY ONE MODEL]

Steady-state visually evoked potential (SSVEP) is an HMI setup which allows machines to detect the focus of a patient based on brain waves. In this setup, the patient wears a brainwave capturing EEG cap and is facing a display that renders UI with different refresh rates (change in frequency). When the patient focuses on a UI element with a particular frequency, it shows as spikes in the Fourier transform of the brain waves.

Assuming an SSVEP setup, develop a time-series machine learning model which determines the refresh frequency of the UI element the user is focused on.

Module 2A: EEG SSVEP Time-Series Recognition

In this setup the model must operate on the data in time-domain.

- Inputs (X's): EEG brain-wave signals (time-domain)
- Outputs (Y's): Frequency band
- Performance:
  - Accuracy (F1-Score): Minimum 70%, typical 75~85%, ideal 95%
  - FPS: Minimum 15 FPS, typical 25~35 FPS, ideal ~60FPS
- Dataset: MAMEM EEG SSVEP Dataset III using Emotive (https://bit.ly/3R4j6DC).
- Resources: Check out the dataset website and their online code repository (e.g. github) for code snippets and utilities to convert and extract the data. The maintainers of the dataset have published code snippets to load and prepare the dataset in Matlab (https://github.com/MAMEM/eeg-processing-toolbox).

Module 2B: EEG SSVEP Frequency Domain Recognition

In this setup convert the data into 64-channels spectrum and use it as the input for your model.

- Inputs (X's): EEG brain-wave signals (64-channels spectrum)
- Outputs (Y's): Frequency band
- Performance:
  - Accuracy (F1-Score): Minimum 70%, typical 75~85%, ideal 95%
  - FPS: Minimum 15 FPS, typical 25~35 FPS, ideal ~60FPS
- Dataset: MAMEM EEG SSVEP Dataset III using Emotive (https://bit.ly/3R4j6DC).
- Resources: Check out the dataset website and their online code repository (e.g. github) for code snippets and utilities to convert and extract the data. The maintainers of the dataset have published code snippets to load and prepare the dataset in Matlab (https://github.com/MAMEM/eeg-processing-toolbox).

Module 3: EEG SSVEP Recognition [IMPLEMENT ONLY ONE MODEL]

Module 3A: Word-Level Predictive Text Module

Develop an ML module to predict the next word given the previously typed words. The module should train from a text dataset and be able to give 3 suggestions for the next word given the typed word sequence.

- Inputs (X’s): Word sequence
- Outputs (Y’s): Top 3 suggestions for next word
- Performance:
  - Accuracy (***NOT*** F1-Score): Minimum 70%, typical 75~85%, ideal 95%
  - FPS: Minimum 15 FPS, typical 25~35 FPS, ideal ~60FPS
- Dataset: Project Gutenberg (https://www.gutenberg.org/cache/epub/766/pg766.txt)

Module 3B: Character-Level Predictive Text Module

Develop an ML module to predict the next word given the previously typed characters. The module should train from a text dataset and be able to give 3 suggestions for the next word given the typed character sequence.

- Inputs (X’s): Character sequence
- Outputs (Y’s): Top 3 suggestions for next word
- Performance:
  - Accuracy (***NOT*** F1-Score): Minimum 70%, typical 75~85%, ideal 95%
  - FPS: Minimum 15 FPS, typical 25~35 FPS, ideal ~60FPS
- Dataset: Project Gutenberg (https://www.gutenberg.org/cache/epub/766/pg766.txt)

Assessment Length

Jupyter notebook (code), 8-10 pages, 10 minutes video presentation

Submission notes

Jupyter notebook (code), 8-10 pages, 10 minutes video presentation

Assignment submission Turnitin type

Not Applicable

General Assessment Information

Grading Basis

Standard

Course Schedule

Teaching Week/Module	Activity Type	Content
Week 1 : 15 July - 19 July	Lecture	Introduction to Machine Learning - 1
	Lecture	Introduction to Machine Learning - 2
	Lecture	Supervised Learning: Regression
	Tutorial	Math Refresher • Linear algebra • Calculus
	Laboratory	Introduction to Machine Learning in Python • Python IDE's Jupyter Spyder • Mathematical Libraries NumPy SymPy Pandas SciPy • Plotting and Visualisation Matplotlib Seaborne
Week 2 : 22 July - 26 July	Lecture	Supervised Learning: Classification
	Lecture	Model Validation
	Lecture	Data Exploration
	Tutorial	Problem Formulation
	Laboratory	Introduction to Machine Learning in Python • Scikit Learn • Plotting and Visualisation Matplotlib Seaborne
Week 3 : 29 July - 2 August	Lecture	Data Engineering
	Lecture	Feature Extraction and Selection
	Lecture	Dimensionality Reduction
	Tutorial	Feature Selection
	Laboratory	Feature Selection
Week 4 : 5 August - 9 August	Lecture	Decision Trees
	Lecture	Ensemble Machine Leaning Models
	Lecture	Association Rules
	Tutorial	Problem Solving: Decision Trees
	Laboratory	Decision Trees and Forests
Week 5 : 12 August - 16 August	Lecture	Neural Networks
	Lecture	Feature Learning via Deep Neural Networks
	Tutorial	Support Vector Machines
	Laboratory	Introduction to Tensorflow
Week 6 : 19 August - 23 August	Lecture	Unsupervised Learning - 1
	Lecture	Unsupervised Learning - 1
	Lecture	Unsupervised Learning - 1
	Tutorial	Problem Solving: Unsupervised Learning
	Laboratory	Unsupervised Learning
Week 7 : 9 September - 13 September	Lecture	Machine Learning w/ Image Data - 1
	Lecture	Machine Learning w/ Image Data - 2
	Lecture	Machine Learning w/ Image Data - 3
	Tutorial	Image Data: Local Structural Similarity
	Laboratory	Convolutional Neural Networks with Tensorflow
Week 8 : 16 September - 20 September	Lecture	Handling Incomplete Data
	Lecture	Handling Incomplete Data - 2
	Laboratory	Handling Incomplete Data Students to work on the project.
Week 9 : 23 September - 27 September	Lecture	Machine Learning w/ Time-Series Data - 1

	Lecture	Machine Learning w/ Time-Series Data - 2
	Lecture	Machine Learning w/ Time-Series Data - 3
	Tutorial	Discussion on Data Collection
	Laboratory	Recurrent Neural Networks with Tensorflow
Week 10 : 30 September - 4 October	Lecture	Machine Learning w/ Natural Language Data - 1
	Lecture	Machine Learning w/ Natural Language Data - 2
	Lecture	Machine Learning w/ Natural Language Data - 3
	Tutorial	Preparing natural language datasets
	Laboratory	• Natural Language Processing with NLTK • Natural Language Processing with Tensorflow
Week 11 : 7 October - 11 October	Lecture	MILITARY TRAINING DAY
	Lecture	Generative Machine Learning Model - 1
	Lecture	Generative Machine Learning Model - 2
	Tutorial	Training with Synthetic Data
	Laboratory	MILITARY TRAINING DAY
Week 12 : 14 October - 18 October	Lecture	Deep Reinforcement Learning - 1
	Lecture	Deep Reinforcement Learning - 2
	Lecture	Self-Supervised Learning
	Tutorial	Training with Synthetic Data
	Laboratory	Triplet Neural Network
Week 13 : 21 October - 25 October	Lecture	Machine Learning with 3D Data
	Lecture	Machine Learning with 3D Data
	Lecture	Ethics and Safety
	Tutorial	Training with Synthetic Data
	Laboratory	Triplet Neural Network

Attendance Requirements

Students are strongly encouraged to attend all classes and review lecture recordings.

Course Resources

Recommended Resources

Probabilistic Machine Learning, 1st Edition, 2022, Kevin Murphy, MIT Press · ISBN 9780262369305.

Course Evaluation and Development

MyExperience Survey

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Convenor	Mo Hossny		B36 Room 103	0251145363		Yes	Yes
Lecturer	Heba El-Fiqi		B15 Room 204	025114 5332		Yes	No

Other Useful Information

School-specific Information

The Leaning Management System

Moodle is the Learning Management System used at UNSW Canberra. All courses have a Moodle site which will become available to students at least one week before the start of semester. Please find all help and documentation (including Blackboard Collaborate) at the Moodle Support page.

UNSW Moodle supports the following web browsers:
- Google Chrome 50+
- Safari 10+

Internet Explorer is not recommended. Addons and Toolbars can affect any browser's performance.

Operating systems recommended are:
- Windows 10,
- Mac OSX Sierra,
- iPad IOS10

Further details:
[Moodle System Requirements]
[Moodle Log In]

If you need further assistance with Moodle:

For enrolment and login issues please contact:
IT Service Centre
Email: itservicecentre@unsw.edu.au
Phone: (02) 9385-1333
International: +61 2 9385 1333

For all other Moodle issues please contact:
External TELT Support
Email: externalteltsupport@unsw.edu.au

Phone: (02) 9385-3331

International: +61 2 938 53331

Opening hours:

Monday – Friday 7:30am – 9:30 pm

Saturday & Sunday 8:30 am – 4:30pm

Study at UNSW Canberra

Study at UNSW Canberra has lots of useful information regarding:

- Where to get help
- Administrative matters
- Getting your passwords set up
- How to log on to Moodle
- Accessing the Library and other areas.

UNSW Canberra Student Hub

For News and Notices, Student Services and Support, Campus Comminity, Quick Links, Important Dates and Upcoming Events

School Contact Information

Deputy Head of School (Education): Dr Erandi Hene Kankanamge

E: e.henekankanamge@adfa.edu.au

T: 02 5114 5157

Syscom Admin Support: syscom@unsw.edu.au

T: 02 5114 5284

Syscom Admin Office: Building 15, Level 1, Room 101 (open 10am to 4pm, Mon to Fri)