IML Course Details

- Lectures
  - Powerpoint presentation and pdfs

- Labs
  - Jupyter Notebook
  - Sections
    - *Introduction:* briefly introduce the purpose of the lab, list the learning outcomes, explain how does the topic fit in overall scheme of the course, and describe the importance of the topic from industry point of view
    - Sections as per the instructions given below
    - Involve students into discussions where necessary
    - Leave some (only small) parts empty and ask students to complete
    - Share the complete notebook at the end of the lab

- Weekly-Self-Practice
  - A small task, properly explained, that students will solve to expand their understanding of the week's topic

- Assignments (70%)
  - Assignment 1 (2 weeks) 30%
  - Assignment 2 (3 weeks) 40%

- Oral Midterm Exam (30%)
  - based on Self-practice (so self-practice gets indirectly counted)

- Bonus Task (20%)
  - An interesting and challenging task to be solved in a week's time

- Participation in Labs and Lectures (5%)

—-

Before the start of the semester, students must install Anaconda on their laptops, and get familiar with Jupyter Notebooks, and Google Colab

—-

Week 1:

First Lab (Crash course on Python) — Variables and Datatypes, Operators (Arithmetic, Logical, Comparison, Assignment, Membership, Conditional Statements, Iteration (for and while loops), Functions, Objects and Classes, and Python Libraries for ML (Numpy, Matplotlib, Seaborn, Pandas, and Scikit-Learn

Self-practice: a small task on visualising data

Week 2:

Second Lab (Regression Analysis) — Reading and viewing data from csv/seaborn, Separating features and labels, Converting categorical data to numbers, Separating training and test sets, Data scaling and Normalisation, Implementing Linear Regression with sklearn, Interpreting the results of regression, strength of LR, weakness of LR, Implementing Polynomial Regression with sklearn

Self-practice: A small task to understand underfitting and overfitting in LR and PR

Week 3:

Third Lab (Logistic Regression) — Comparing classification problem (data) with regression problem (data), Manual Feature Selection (spotting irrelevant features and dropping them), Usual and necessary data pre-processing and preparation, Implementing Logistic Regression with sklearn, Interpreting the results of logistic regression (accuracy, confusion matrix, recall, precision, f1-score), strength/weakness of logistic regression, scaling to multi-class problems

Self-practice: A small task to understand the impact of class imbalance

Week 4:

Fourth Lab (NB, KNN, Regularisation) — A dataset (problem) that shows the importance/need of NB classifier, implementing NB with sklearn, interpreting the results of NB, A dataset (problem) that shows the importance/need of KNN classifier, implementing KNN with sklearn, interpreting the results of KNN, Using GS to tune the value of K, Going back to regression (implementing ridge and lasso regression with sklearn)

Self-practice: A small task to compare L2 with L1 regulariser

Week 5:

Fifth Lab (Dimensionality Reduction) — A dataset that highlights the presence of correlated features, the problem of overfitting, and the need for dimensionality reduction, standardising the data, implementing PCA with sklearn, analysing/interpreting the result of PCA, choosing the number of PCs, going back to regularisation (When to use Lasso? When to use Ridge? How we can decide which regularisation? and decide the value of lambda?

Self-practice: A small task to understand how to compute covariance matrix from data, and how to compute eigen-vectors and eigen-values

Week 6:

Sixth Lab (SVM) — Generate a linear separable binary data, visualise the data, draw three random separating hyperplanes and discuss which one is better, classify the data with linear SVC, plot SVC decision boundary and analyse support vectors, show the sensitivity of hard margin sum to outliers, show the impact of hyper-parameter c, Generate a non-linear binary data, visualise it and discuss why linear SVC won’t yield proper results, classify the data with kernel SVM

Self-practice: A small task to see how multi-class problems can be solved with SVM

Week 7:
Seventh Lab (ANNs) — Solve one epoch of backprop on board for a small ANN, Introduction to TensorFlow and Keras, explaining and demonstrating the role of Model, Sequential, Dense and optimiser (Adam), epochs, batch size, learning rate, etc, Implementing a densely connected neural network with TensorFlow for two types of data (Images, Timeseries, etc.), plotting and using the train/test losses, etc.

Self-practice: A small task to see how ANNs can also be used for regression tasks.

Week 8:
Oral Midterm Week: meeting with every student, where she/he demonstrates her/his homeworks and answers 2-3 questions related to their homeworks, lectures and labs.

Week 9:
Ninth Lab (CNNs) — show what is convolution and how can it be implemented in python (convolve an image with a filter to detect vertical and horizontal edges/lines), explain Conv2D (and its inputs), explain MaxPool2D, explain Flatten, Introduce a problem (data) where CNNs would work better than feed-forward ANNs (explain why), implement a CNN to solve this task,

Self-practice: A small task to understand the impact of number of layers in CNN, padding, strides, pooling (max vs average, etc.)

Week 10:
Tenth Lab (Techniques for training DNNs) — take the previous weeks' tasks and couple them with the following to show how they can be implemented and what can be their impact: Dropout, Data Augmentation, Batch Normalisation, Layer Normalisation, Learning Rate Scheduling, Gradient Clipping, Early Stopping, etc.)

Self-practice: A small task to understand what are pertained models and how can one use them

Week 11:
Eleventh Lab (DTs) — Introduce a problem (data) where inferencing would be important but hard to achieve using other classifiers like ANNs, Train a Decision Tree with sklearn for this problem, Visualise the learned tree, Show decision tree boundaries, demonstrate the sensitivity of the tree to the training set details, show how to learn better trees using pruning and other similar techniques,

Self-practice: A small task to understand how DT can be used for regression

Week 12:
Twelfth th Lab (Ensemble Learning, RFs and Boosting) — Introduce a problem (data) where RF would perform better than a DT - explain why, Solve the problem using both a single tree and a RF, compare the decision boundaries of both and discuss with students, compare the decision boundaries of each individual tree in the RF with that of RF to show that even though each individual might do a bad job but the ensemble (RF) overall does a good job - discuss/ask them why, Solve the same problem with AdaBoost

Self-practice: A short task to understand how different models (RF, Linear SVC, and MLP) can be trained and combined into an ensemble using VotingClassifier.

Week 13:

Thirteenth Lab (Clustering 1) — generate dummy (2D) data using make_blobs and plot it, cluster the data using k-means with sklearn, plot the result (along with cluster centres) and analyse it, study the impact of having more or less number of clusters, Introduce a real-world unsupervised learning clustering problem, discuss the importance of solving this clustering problem, solve the problem using k-means with sklearn-identify the optimum number of clusters manually, demonstrate k-means variability with respect to initial cluster centres (by setting init = random), show different problems (data) where k-means might not work well, discuss why

Self-practice: a short task to learn how one can determine the optimum number of clusters using the elbow method.

Week 14:

Fourteenth Lab (Clustering 2) — Introduce various problems (datasets) where k-means might not work well, explain AgglomerativeClustering class, use it to cluster the data, plot and discuss the results, Trying different types of linkage algorithms in Agglomerative Hierarchical Clustering, explain scipy.cluster.hierarchy module and show how can it be used to draw dendrograms, Introduce a clustering problem having outliers, explain the DBSCAN class, use it solve the clustering problem, analyse the impact of eps and n_neighbors.

Self-practice: A small task to see how clustering can be used for image segmentation.

Week 15:

Bonus Task Implementation
Self-reflection using carefully designed (1 Grade Point for attendance) 5-6 questions

Assignment 1
Open: start of Week 5
Close: end of Week 6
Grading and Feedback: end of Week 7

Assignment 2
Open: start of Week 10
Close: end of Week 12
Grading and Feedback: end of Week 14

Bonus Task
Open: start of Week 15
Close: end of Week 15
Grading and Feedback: end of Week 16