ELEC70143 - Machine Learning Assignment 2025/2026

This is a two-week assignment (~20 hours expected effort). Focus on reasoning, interpretation, and clear figures rather than extensive code development.
Many of the questions are open-ended: there will be multiple valid approaches to them. Marks will be allocated based on the individual merits of the approach you adopt, rather than the correct choice of any specific strategy.

Submission: a PDF file with clear section headings matching the question numbers. Figures and Tables must be clearly explained and interpretable. You may use a Jupyter notebook, converted to PDF at the end.

The submission deadline is December 2nd at Midday .

This is an individual project: you must produce your own work; in line with your library services and plagiarism induction received at the start of term. See the Teams General Files for the presentations to remind yourself or refer back to your online plagiarism awareness course.

Part 1: Air Pollution Monitoring System

Air pollution remains one of the leading environmental causes of premature mortality, responsible for an estimated **7 million deaths per year worldwide** according to the WHO. Measuring pollutant concentrations accurately in urban areas is essential for managing air quality and protecting public health. However, **deploying accurate monitoring systems is expensive**: reference-grade sensors for carbon monoxide (CO), nitrogen oxides, and particulate matter can cost thousands of pounds each, while lower-cost sensors provide noisier signals.

To support municipal decision-making, you are working with a start-up that wants to **design a compact, low-cost air-quality monitor**. The monitor uses a small set of electrochemical and meteorological sensors to estimate the **true concentration of carbon monoxide** in the atmosphere. Because of space and cost constraints, not all sensors can be included in the final design.

Your goal is to:

1. Build and compare regression models that predict CO concentration from other sensor readings and environmental variables.
2. Quantify the trade-off between **model accuracy** and **hardware cost**—that is, determine which subset of sensors should be purchased given a fixed budget.
3. Advise decision-makers on the most cost-effective monitoring configuration.

As a testbed, we shall use the data from the paper De Vito, S., Massera, E., Piga, M., Martinotto, L., & Di Francia, G. (2008). On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario, collected in an Italian city in 2004–2005. Each row corresponds to a snapshot of ambient conditions and sensor responses. The target variable is **CO(GT)** — the true CO concentration measured in mg/m^3. The remaining variables are features.

The data can be found in q1.csv

A summary of the variables is shown below.

Variable	Description	Typical range
PT08.S1(CO)	Tin oxide sensor response (ppm)	300–1800
PT08.S2(NMHC)	Non-methane hydrocarbon sensor	400–2000
PT08.S3(NOx)	Nitrogen oxide sensor	400–2000
PT08.S4(NO2)	Nitrogen dioxide sensor	400–2000

PT08.S5(O3)	Ozone sensor	400–2000
T	Temperature (°C)	–10 – 40
RH	Relative humidity (%)	0 – 100
AH	Absolute humidity	0 – 50
NOx(GT)	Reference NO_x concentration	0 – 600
NO2(GT)	Reference NO_2 concentration	0 – 250
C6H6(GT)	Benzene concentration (µg/m^3)	0–15
CO(GT)	Ground-truth CO concentration (target)	0 – 10

A. After you import the data into Python, you will notice that there are several missing values, marked as NaN. For each feature, report the percentage of missing entries. Propose a strategy for handling the missing data and implement it. Hint: You may use mean/median imputation, k-nearest-neighbour imputation, or model-based imputation, but justify your choice.

B. Using a correlation plot or pairs plot, study the distributions of the main features and the target. Comment on any presence of strong correlations and potential multicollinearity.

C. Split the data randomly into a training set (80%) and test set (20%). Fit an **ordinary least squares** model with all the features. Provide a model summary and list significant variables. Can you draw any conclusions from this? Also, notice that some coefficients are extremely large in magnitude or have very large p-values. Why is this happening? **Hint: I would recommend using the statsmodels OLS function which provides richer information.**

D. Fit **ridge** and **lasso** regressions using cross-validation to tune the regularisation parameter. Plot coefficient paths as a function of the regularisation parameter. Comment on the effect of regularisation on multicollinearity and model interpretability. Plot the regularisation parameter on log-scale and always fit on **scaled** features.

E. Using a polynomial kernel (Degree = 2), fit a kernel ridge regression model. Does this improve predictive performance? Why or why not?

F. Fit a kernel ridge regression with an RBF kernel. Tune both the regularisation parameter and kernel width using grid search and cross-validation. **Note: this can be computationally intensive. Reduce the number of points in the grid-search if it takes too long (i.e. more than a few minutes).**

G. For the above models (OLS; Ridge; Lasso; Kernel Ridge regression with degree-2 polynomial kernel; Kernel Ridge regression with RBF**)** report 6 performance metrics: the in-sample R2, MAE and RMSE and the out-of-sample R2, MAE and RMSE (evaluated on the test set). Compare and contrast the pros and cons of each model in terms of performance and interpretability.

H. Each sensor has a fixed purchase cost.

Feature	**Cost (£)**
PT08.S1(CO)	1,000
PT08.S2(NMHC)	800
PT08.S3(NOx)	700
PT08.S4(NO2)	700
PT08.S5(O3)	900
T	150
RH	150
AH	200
NOx(GT)	1,800
NO2(GT)	1,800
C6H6(GT)	2,500

A portable monitor can include sensors whose total cost does not exceed **£4000**. Using your **lasso** model, construct a sequence of models corresponding to increasing total sensor cost. For each model, record:

- The set of selected features,
- The total cost, and
- The cross-validated RMSE on the training data.

Plot the **Cost vs. RMSE** to visualise the cost–accuracy frontier. To do this: Use a **lasso path** over a grid of regularisation parameters (e.g., 10^{-6} to 1). For each one do 5-fold CV RMSE, identify selected set, and compute total cost.

Identify two feasible designs:

- A **low-cost** monitor (≤ £2500)
- A **high-performance** monitor (≤ £4000).

Evaluate their test-set RMSEs.

I. Urban authorities issue public health alerts when CO exceeds the legal threshold y=5 mg/m^3. The **cost of a false positive** (unnecessary alert) is £2,000 and the **cost of a false negative** (missed dangerous pollution) is £10,000. We decide to adopt the following alert rule.

Deterministic alert rule: For a threshold t on the model's predicted concentration *y,* issue an alert if y>t.

For an appropriate range of thresholds *t,* compute the **expected cost at threshold** t via cross-validation (or on a separate validation set). Based on this, select a good candidate for the threshold.

J. Compare the **expected cost** between the low-cost and high-performance monitor designs. Discuss the relative differences. Are the added sensors justified by the reduction in expected cost?

Part 2: Detecting Cyber Attacks from Network Flows

The UK National Cyber Security Centre (NCSC) must detect malicious traffic occurring over web-based national infrastructure. One of the approaches is doing statistic modelling of the network traffic. Cyber operators will monitor various network statistics over time and identify patterns which are not typical network behaviour and flag them, so that operative teams can investigate possible network incursions.

Generally, it is better to err on the side of caution: False negatives (missed attacks) are very costly. False positives waste analyst time, which is inconvenient but not destructive. For the sake of this assignment, assume that the expected cost of a False Negative is £1,000,000 while the cost of a False Positive is FP=£1,000.

The goal of this question is to develop effective models for classifying the presence of potentially malicious internet traffic. Based on these models, we need to tune them to identify the best decision threshold to minimise the expected cost.

As a test bed, we will use real network traffic data. The supplied dataset contains flow-level records of benign traffic and multiple types of DDoS (Distributed Denial of Service) attacks. Each record includes features such as average packet size, number of distinct destination IPs, variance of inter-packet times, flow duration, and protocol. Each record is labelled as BENIGN or MALICIOUS.

If you're interested in the use of statistical and machine learning methods for cyber-security, you can read Heard, N. A., Adams, N. M., Rubin-Delanchy, P., & Turcotte, M. (Eds.). (2018). **Data science for cyber-security** (Vol. 3). World Scientific.

The data can be found in q2.csv.

A summary of the variables is shown below.

Name	Type
Flow Duration	Continuous
Total Fwd Packets	Continuous
Total Backward packets	Continuous
Total length of Fwd packets	Continuous
Total length of Bwd packets	Continuous
Flow IAT mean	Continuous
Flow IAT std	Continuous
Packet Length mean	Continuous

Packet length Std	Continuous
Average Packet Size	Continuous
Active Mean	Continuous
Idle Mean	Continuous
Protocol	Categorical (6, 17, 0).
Target	1/0

General hint: When fitting the models, it makes sense to apply a **one-hot encoding** to the Categorical variable (Protocol). This transforms protocol into 3 separate columns (6, 17, 0) taking values in 0, 1. You can either do this manually, or alternatively, use **OneHotEncoder** in sklearn.

Throughout you should split the dataset into 80% training and 20% test sets, stratified by class label. Hyperparameters should be found using K-fold cross-validation (where the K is of your choosing).

A. Exploratory analysis [10 pts]

Load the data in python.

- Report the class balance.
- Plot distributions of features **Flow Duration** and **Total Fwd Packets** for benign vs attack flows.
  Report two insights about separation between classes.

B. Baseline logistic regression [10 pts]

Fit a logistic regression model. Plot a confusion matrix. Report the accuracy, and ROC, ROC-AUC and comment on their interpretation.

C. Penalised logistic regression [15 pts]

Fit an L1-regularised logistic regression. Report the accuracy and ROC and ROC-AUC. Perform 20 bootstrap resamples of the data and plot how often each feature is selected by measuring the proportion of resamples in which the feature is selected. Visualise and interpret the top 5 most influential features (largest absolute coefficients). Comment briefly (2–3 sentences) on what these imply about benign vs malicious traffic.

D. Linear SVM [15 pts]

Fit Linear SVM with **both** hard and soft margins. In the soft-margin case, the regularisation parameter C should be tuned by CV using grid search over $C \in \{0.1, 1, 10, 100\}$. For each model:

- Report the accuracy, ROC-AUC, as well as the number of support vectors.
- Plot decision boundary in a 2D projection (choose two continuous features).
- Explain why hard margin SVM is fragile in this context.

E. Kernel SVM [20 pts]

Fit an RBF-kernel SVM.

- Tune (C, γ) using cross-validation, using a grid search with $C \in \{\ 0.1, 1, 10, 100\ \}$ and $\gamma \in \{0.001, 0.1, 1, 10\}$.
- Compare test performance to logistic regression.

F. Cost-sensitive scenario [15 pts]

- For each model, evaluate and report the total expected cost using the default decision threshold.
- For each model, adjust the decision threshold to minimise cost (rather than the naive 0.5 threshold).
- Recompute expected costs for all models, and compare, plotting ROC and cost curves to justify your threshold choice.

Hint: To calculate the total expected cost, compute the ROC curve using **sklearn.metrics.roc_curve**, then

$$Cost(t) = C_{FN} \times \left(1 - TPR(t)\right) \times N_{+} + C_{FP} \times FPR(t) \times N_{-},$$

where N_+ and N_- are the positive and negative and positive tests, respectively. Based on this, choose the threshold t that minimises this cost.

G. Reflection [15 pts]

Write a **policy briefing note** for NCSC leadership:

- Compare Logistic, Penalised Logistic, Linear SVM, and Kernel SVM across:
  - Interpretability
  - Accuracy
  - Stability under resampling
  - Robustness to changing costs
- Recommend a **deployment strategy** balancing explainability and detection power (e.g., interpretable model for baseline monitoring; kernel SVM as escalation filter).
- Discuss risks: what if new attack types shift feature distributions? Which models are most vulnerable, and how could monitoring/adaptation mitigate these risks?

Length for question G: max 500 words, covering (1) Comparison, (2) Deployment recommendation, (3) Risk and adaptation.