AMATH 445/645 Assignment 1 Winter 2026

Due date: Monday, February 2 at 4:30 pm

Note: Undergraduate students are not required to do Q2 but they're encouraged to attempt it.

Question 1: Supervised Learning for Bioprinting

Background: Bioprinting uses additive manufacturing to create three-dimensional structures containing living cells, enabling controlled *in vitro* experiments that better mimic the tumour microenvironment. Certain printing parameters, such as extrusion pressure and nozzle diameter, can substantially reduce post-printing cell viability. Recent work has shown that supervised machine learning models can predict printing outcomes, including cell viability, from printer and bioink parameters. In *Machine Assisted Experimentation of Extrusion-Based Bioprinting Systems*, Tian et al. [1] compiled a multi-laboratory dataset of 617 cell viability measurements and applied several ML models to predict printing outcomes.

Objective: In this question, you will partially reproduce and extend the supervised learning analysis of Tian et al. [1]. You will preprocess the experimental dataset and train two classification models to predict whether a given set of bioprinting parameters results in *acceptable* cell viability (Yes/No).

Assume that cell viability is considered *acceptable* if it exceeds the threshold defined in the paper.

(1a) Written Long Answer

1. Define and explain **precision** and **recall** in the context of binary classification. Provide and explain the formula for each metric.
2. Why can accuracy alone be misleading when evaluating classification models? Explain why precision and recall are important complementary metrics, and give a concrete example.

(1b) Data Preprocessing

Download the dataset from https://osf.io/qd5k8. Preprocess the data following Section 2.3 (Data Preprocessing) of the paper. For this assignment, **skip the feature selection step**.

Your preprocessing pipeline should include the following steps:

- Handling missing values for bioink temperature as described in the paper
- Removing features with more than 50% missing or zero values
- Imputing remaining missing values using `KNNImputer` from `scikit-learn`
- Scaling continuous features using `MinMaxScaler`
- Removing "Acceptable Pressure" from the feature set

You may upload the dataset directly to your Google Colab session and use libraries such as `pandas` and `scikit-learn` for preprocessing.

Submit your preprocessing code.

(1c) Decision Tree Classifier

Train a Decision Tree classifier on the preprocessed data using default `scikit-learn` parameters. Submit a PDF of your code and discuss the model's performance using **accuracy, precision, and recall**.

(1d) Support Vector Machine

Train a Support Vector Machine (SVM) classifier on the same data using default `scikit-learn` parameters. Submit a PDF of your code and discuss the model's performance using **accuracy, precision, and recall**.

Briefly compare the performance of the Decision Tree and SVM models.

Question 2: Classification of Phases in the Ising Model

Note: This question is intended for graduate students. Undergraduate students are encouraged to attempt it as an optional challenge.

Background: The Ising model is a fundamental model in statistical mechanics consisting of binary spin variables ($+1$ or -1) arranged on a lattice. Despite its simplicity, the model exhibits a phase transition: as temperature increases, the system transitions from an *ordered phase*, characterized by large domains of aligned spins, to a *disordered phase* in which spin orientations are largely random.

Objective: In this question, you will generate spin configurations of the two-dimensional Ising model using Monte Carlo sampling and use supervised learning to classify these configurations as belonging to either the ordered or disordered phase. The goal is to investigate how well a simple linear classifier can detect collective physical behavior and phase transitions from high-dimensional data.

Data Generation using Monte Carlo

- Follow the algorithm described in this tutorial to generate Ising model configurations using the Monte Carlo method.
- Generate configurations over a range of temperatures, with particular emphasis on values below, near, and above the critical temperature
$$T_c \approx 2.269$$
for a two-dimensional square lattice.
- You may use the provided (untested) Python implementation available in this Google Colab notebook, or submit your own implementation.
- Use a fixed lattice size (e.g. $L = 20$). Flatten each $L \times L$ lattice configuration into a one-dimensional feature vector with L^2 features.

Labeling

- Label configurations generated at temperatures $T < T_c$ as *ordered.*
- Label configurations generated at temperatures $T > T_c$ as *disordered.*
- Note that configurations near T_c may be intrinsically difficult to classify due to critical fluctuations.

Data Preparation

- Shuffle the dataset and split it into training (80%) and test (20%) sets.
- Standardize the features prior to training.

(2a) Logistic Regression

- Train a Logistic Regression classifier on the training data using default `scikit-learn` parameters.

- Submit a PDF of your code.
- Evaluate performance on the test set using accuracy, precision, recall, and F1-score.

(2b) Written Long Answer

1. Describe how Monte Carlo spin configurations differ qualitatively at temperatures far below T_c, near T_c, and far above T_c.
2. How well does Logistic Regression classify the phases of the Ising model using these configurations? Discuss any limitations you observe, particularly near the critical temperature.

Notes: Ensure that the dataset is shuffled before splitting. Configurations generated at similar temperatures may be highly correlated, and shuffling helps prevent biased training and evaluation.

Question 3: Logistic Regression with Regularization

Consider logistic regression: $P(y = 1 \mid \mathbf{x}) = \sigma(\boldsymbol{\beta}^\top \mathbf{x})$, $\sigma(z) = 1/(1 + e^{-z})$.

a) Why does unregularized LR fail with linearly separable data?
b) With L2 regularization $J(\boldsymbol{\beta}) = -\sum_i [y_i \log \sigma_i + (1 - y_i) \log(1 - \sigma_i)] + \frac{\lambda}{2} \|\boldsymbol{\beta}\|_2^2$, why does $\lambda > 0$ ensure a finite solution?
c) How does increasing λ affect bias and variance?
d) Lowering threshold from 0.5 to 0.3: effect on FPR/FNR? When useful?

Question 4: Support Vector Machines

Support Vector Machines (SVMs) find a separating hyperplane by maximizing the margin. Consider a linear SVM with decision function $f(x) = w^\top x + b$.

1. A linear SVM is trained on 2D data where $x_1 \in [0, 1]$ and $x_2 \in [0, 1000]$. Explain why failing to scale these features can cause the learned decision boundary to be dominated by the x_2 feature.
2. Now suppose the dataset is highly imbalanced, with 1% positive and 99% negative examples. Explain why a standard soft-margin SVM with a single penalty parameter C may perform poorly on the minority (positive) class.
3. Briefly describe one specific modification to the SVM training process or objective function that addresses *either* the feature scaling issue *or* the class imbalance issue.

References

[1] Shuyu Tian et al. “Machine assisted experimentation of extrusion-based bioprinting systems”. In: *Micromachines* 12.7 (2021), p. 780.