# Assignment 3: Non-Linear Models and Validation Metrics (50 marks total)

Due: March 7 at 11:59pm

Name:

In this assignment, you will need to write code that uses non-linear models to perform classification and regression tasks. You will also be asked to describe the process by which you came up with the code. More details can be found below. Please cite any websites or AI tools that you used to help you with this assignment.

## Part 1: Classification (30 marks)

### *Part 1A: Decision Function*

Building on the first part of the previous assignment, we would like to test how changing the decision boundary for a linear model impacts the results.

#### Step 0: Import Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore') #ignoring some deprication warnings
```

#### Step 1: Data Input (1 mark)

Load spam data using the same method as assignment 2.

```
# TO DO: Import spam dataset from yellowbrick library
```

Next, we need to split the data into training and testing sets.

```
# TO DO: Split 10% of the data for the testing set (1 mark)
```

#### Step 2: Data Processing

Based on assignment 2, follow the same data processing steps (if needed)

```
# TO DO: Data processing steps (if needed)
```

## Step 3: Implement Machine Learning Model (2 marks)

1. Import `LogisticRegression` from sklearn
2. Instantiate model `LogisticRegression(max_iter=2000)`
3. Split training data into training and validation sets (use 20% of the data for validation)
4. Train the machine learning model

```
# TO DO: Split training data into training and validation sets and fit model to training data (2 marks)
```

## Step 4-5: Validate Model and Visualize Results (4 marks)

Next, we can print the classification report and confusion matrix for this data set using the training and validation data.

```
# TO DO: Print classification report (1 mark)

# TO DO: Print confusion matrix (1 mark)
```

In this case, do we want to increase precision or recall? Based on your choice, select the appropriate direction to adjust the decision boundary. You can use either 1 or -1 as your new threshold, depending on your selected direction.

```
# TO DO: Predict values based on new decision function threshold and print classification report (1 mark)

# TO DO: Print new confusion matrix (1 mark)
```

## Questions (6 marks)

1. Did you decide to move the threshold to increase recall or precision? Why?
2. How did your decision impact the number of false negatives and false positives? What is a potential unintended consequence of changing the decision boundary in this context?
3. Why did we use the validation data instead of the test data to check the new decision threshold?

*ANSWER HERE*

# *Part 1B: Non-linear classification*

Using the spam dataset from part 1A, compare the performance of two non-linear models to the linear model used in assignment 2.

## Step 3: Implement Machine Learning Models

1. Import `LogisticRegression`, `SVC` and `RandomForestClassifier` from sklearn
2. Instantiate models as `LogisticRegression(max_iter=2000)`, `SVC()` and `RandomForestClassifier(random_state=0, max_depth=10)`

3. Implement the machine learning models using cross-validation (Step 4)

## Step 4: Validate Model

Calculate the training and validation accuracy for the three different models mentioned in Step 3. For this case, you can use `cross_validate()` with `cv=5` and `scoring='accuracy'` to get the training and validation data for each of the three models and calculate the accuracy results.

## Step 5.1: Visualize Results (4 marks)

1. Create a pandas DataFrame `results` with columns: Training accuracy and Validation accuracy
2. Add the training accuracy and validation accuracy for each model to the `results` DataFrame
3. Add the model names as the index for the DataFrame
4. Print `results`

```
# TO DO: ADD YOUR CODE HERE FOR STEPS 3-5
# HINT: USING A LOOP TO STORE THE DATA IN YOUR RESULTS DATAFRAME WILL
BE MORE EFFICIENT
```

SVM is sensitive to feature ranges, so scaling may be needed. Look at the feature ranges and try using a scaling method to see if the SVM results are improved.

```
# TO DO: Look at the ranges for each feature (0.5 marks)
# Hint: there is a built-in pandas function that you can use to view
the statistics of your data

# TO DO: Implement scaling for SVM and print training and validation
accuracies (1.5 marks)
```

Which model gave us the best results? Use that model for the next section.

## Step 5.2: Visualize Classification Errors (3 marks)

In this section, print the classification report and confusion matrix to investigate the recall vs. precision for the best model. Use the full training set and testing set for this analysis.

```
# TO DO: Train model and find predicted values for testing set using
best model (1 mark)

# TO DO: Print classification report (1 mark)

# TO DO: Print confusion matrix using a heatmap (1 mark)
```

## Questions (6 marks)

1. Which model did you select for part 5.2? How did it compare to the other models?

2. Looking at the feature ranges, would a tree-based model or SVM make more sense for this dataset? Did using scaling for SVM improve the results? How did it perform compared to the random forest model?
3. In your opinion, is it better to focus on changing the decision threshold or changing the model to improve precision/recall results? Why?

*ANSWER HERE*

## Process Description (4 marks)

Please describe the process you used to create your code for parts A and B. Cite any websites or generative AI tools used. You can use the following questions as guidance:

1. Where did you source your code?
2. In what order did you complete the steps?
3. If you used generative AI, what prompts did you use? Did you need to modify the code at all? Why or why not?
4. Did you have any challenges? If yes, what were they? If not, what helped you to be successful?

*DESCRIBE YOUR PROCESS HERE - BE SPECIFIC*

# Part 2: Regression (18 marks)

For this section, we will be using the concrete example from yellowbrick. Since this dataset is highly non-linear, we will be evaluating how well different tree-based models work for this case.

You will need to repeat the steps from Part 1 for this analysis.

## Step 1: Data Input (1 mark)

The data used for this task can be imported using the yellowbrick library: https://www.scikit-yb.org/en/latest/api/datasets/concrete.html

Use the yellowbrick function `load_concrete()` to load the concrete dataset into the feature matrix `X` and target vector `y`.

```
# TO DO: Import concrete dataset from yellowbrick library (1 mark)
```

## Step 2: Data Processing (1 mark)

Check if there are any missing values and fill them in if necessary. Remove any non-numeric columns.

```
# TO DO: Process the data - fill-in any missing values and remove any
non-numeric columns (0.5 marks)
```

The concrete data should already be split into the feature matrix and target vector. Inspect the first few columns of the feature matrix.

```
# TO DO: Inspect the first few rows of the feature matrix (0.5 marks)
```

## Step 3: Implement and Validate Machine Learning Model (7 marks)

1. Import any required libraries
2. Split the data into training and testing sets (testing data should be 10% of the dataset)
3. Train and validate the Decision Tree model with the training set (use `cross_validate()` with `cv=5` and `scoring='r2'`)
    a. Test five different max_depth values: 3, 5, 7, 9 and 11
4. Print the training and validation accuracy for the best max_depth results. Which max_depth gave us the best results?

**Note**: for any random state parameters, you can use random_state = 0

```
# Split training and testing sets (1 mark)

# Test max_depths of 3, 5, 7, 9 and 11 for a decision tree model to
find the best results (3 marks)
# Hint: It is easier if you use a loop to evaluate each max_depth
```

Now that we have found the best results for a decision tree model with this dataset, let's compare this result to using `Random_Forest` or `GradientBoosting`. For both models, use `max_depth=5`.

```
# TO DO: Calculate and display training and validation accuracies for
both models using default hyperparameters (3 marks)
```

Which model gave us the best results? Use that model for the next step.

## Step 4: Test Model (1 mark)

Select the best model and calculate the testing accuracy using the R^2 score

```
# TO DO: Find test score using best model (1 mark)
```

## Questions (4 marks)

1. Out of the models you tested, which model would you select for this dataset and why?
2. If you wanted to increase the accuracy of the tree-based models, what would you do? Provide two suggestions.

*ANSWER HERE*

## Process Description (4 marks)

Please describe the process you used to create your code. Cite any websites or generative AI tools used. You can use the following questions as guidance:

1. Where did you source your code?

2. In what order did you complete the steps?
3. If you used generative AI, what prompts did you use? Did you need to modify the code at all? Why or why not?
4. Did you have any challenges? If yes, what were they? If not, what helped you to be successful?

*DESCRIBE YOUR PROCESS HERE - BE SPECIFIC*

# Part 3: Reflection (2 marks)

Include a sentence or two about:

- what you liked or disliked,
- found interesting, confusing, challenging, motivating while working on this assignment.

*ADD YOUR THOUGHTS HERE*