# Project 2 COE 347L

Mihiro Suzuki
MS84579

# Data Preparation

## Looking at raw data

By observing the raw dataset I saw that there are some missing values that have “?” values in it. This showed me that I will need to replace it with some sort of value.

## Data size

By running simulations, I found that the raw dataset consists of 286 rows and 10 columns, making its total size 2,860 data points (row x columns) .

## Missing data

In order to handle the missing data, I first found the missing data points. Since the 2 missing data columns were 'node-caps', 'breast-quad' which had categorical outputs. Therefore, I used the mode to plug in an “average” value.

## One hot encoding

To allow for a model to be trained on this dataset, I replaced the categorical values with a true false column. This creates a numerical category that allows for numerical relationships to be drawn.

# Model Training Techniques and Performance

1. **KNN**

The first technique that I used to train the model was the K Nearest Neighbor. I first found the optimum K to use according to the accuracy value, then implemented the KNN model with said value.

```
Best k value: {'n_neighbors': 21}
Best score: 0.735
```

This model is good for precision but it should optimize for recall since the cost of false negatives are high in this context. This is because it is better for someone to be

warned that they may have recurrence, and not, rather than not be warned and have a recurrence. Therefore, I re optimized the k for recall rather than accuracy. This increased the recall from 0.19 -> 0.27.

```
Performance on TEST
********************
              precision    recall  f1-score   support

           0       0.74      1.00      0.85        60
           1       1.00      0.19      0.32        26

    accuracy                           0.76        86
   macro avg       0.87      0.60      0.59        86
weighted avg       0.82      0.76      0.69        86

Performance on TRAIN
**********************
              precision    recall  f1-score   support

           0       0.75      0.98      0.85       141
           1       0.81      0.22      0.35        59

    accuracy                           0.76       200
   macro avg       0.78      0.60      0.60       200
weighted avg       0.77      0.76      0.70       200
```

->

```
Performance on TEST
********************
              precision    recall  f1-score   support

           0       0.72      0.80      0.76        60
           1       0.37      0.27      0.31        26

    accuracy                           0.64        86
   macro avg       0.54      0.53      0.53        86
weighted avg       0.61      0.64      0.62        86

Performance on TRAIN
**********************
              precision    recall  f1-score   support

           0       0.85      0.99      0.91       141
           1       0.94      0.58      0.72        59

    accuracy                           0.86       200
   macro avg       0.90      0.78      0.81       200
weighted avg       0.88      0.86      0.85       200
```

## 2. Logistic Regression

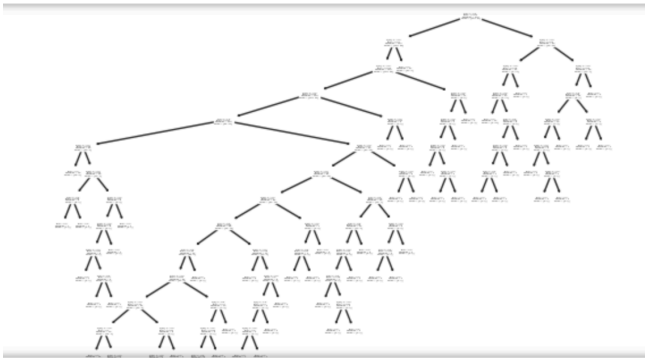
The logistic regression was also split into the same training and testing. The accuracy is a bit lower but the recall value was higher

```
Performance on TEST
********************
              precision    recall  f1-score   support

           0       0.74      0.87      0.80        60
           1       0.50      0.31      0.38        26

    accuracy                           0.70        86
   macro avg       0.62      0.59      0.59        86
weighted avg       0.67      0.70      0.67        86

Performance on TRAIN
**********************
              precision    recall  f1-score   support

           0       0.78      0.91      0.84       141
           1       0.66      0.39      0.49        59

    accuracy                           0.76       200
   macro avg       0.72      0.65      0.67       200
weighted avg       0.75      0.76      0.74       200
```

## 3. Decision Tree

The decision tree seems to be overfitted. This is obvious from the precision value in the training dataset and the tree diagram.

```
Performance on TEST
********************
              precision    recall  f1-score   support

           0       0.72      0.80      0.76        60
           1       0.37      0.27      0.31        26

    accuracy                           0.64        86
   macro avg       0.54      0.53      0.53        86
weighted avg       0.61      0.64      0.62        86

Performance on TRAIN
**********************
              precision    recall  f1-score   support

           0       0.96      1.00      0.98       141
           1       1.00      0.90      0.95        59

    accuracy                           0.97       200
   macro avg       0.98      0.95      0.96       200
weighted avg       0.97      0.97      0.97       200
```



## 4. Random Forest

The random forest method has a low accuracy score but has a very high recall score. The depth was a bit large likely causing a bit of overfitting.

```
Performance on TEST
********************
              precision    recall  f1-score   support

           0       1.00      0.02      0.03        60
           1       0.31      1.00      0.47        26

    accuracy                           0.31        86
   macro avg       0.65      0.51      0.25        86
weighted avg       0.79      0.31      0.16        86

Performance on TRAIN
********************
              precision    recall  f1-score   support

           0       0.88      0.05      0.09       141
           1       0.30      0.98      0.46        59

    accuracy                           0.33       200
   macro avg       0.59      0.52      0.28       200
weighted avg       0.71      0.33      0.20       200
```

## Recommended Model

My recommendation for the model depends on the context. For example, if it's better to sacrifice on accuracy and always find people that could potentially have recurrence, I would recommend the random forest method.

## False Positives and False Negatives

The most important metric in my opinion is the recall value. This is because it is better for someone to be warned that they may have recurrence, and not, rather than not be warned and have a recurrence. On the other hand, if the test is super expensive, I would choose a model with higher accuracy.

## ChatGPT

I used ChatGPT to figure out what type of graphs to make and the code for it. It gave me an intuition on where to look. I also used it to comment on my code and debug issues that I ran into in the terminal.