What did you do to prepare the data?

To prepare the data I went through a couple steps:

- First printing the data to take a look at the general contents and structure.
- Following that i printed the data types of the variables to understand the actual data structure component of the data
- Duplicates in the data were dropped
- Next I looked at the info to understand the general shape of the data, along with identifying null and missing values
- Originally I didn't have this step in the preparation but after discovering more problematic data values I added this step to my data preparation. That step is looking at the unique values of all variables to take a glance at the general expected or grossly outlying values.
- In a new data frame I remove null and irregular values and then encode the columns which had these values
- After encoding I then look grouped to find the best value to impute the missing data in the original data frame
- After encoding the data I make a few plots to understand the spread of different variables and the correlations between the different variables.

What insights did you get from your data preparation?

I think the most important insights I got from my data preparation were:

- Simplifying replying on functions for dealing with duplicates, irregular, and null values is not enough and checks and balances are needed to ensure the data has been thoroughly preprocessed
- From the data I also learned the irradiat variable is so heavily leaning no that even the yes response is an outlier
- I also saw the age distribution being unimodal primarily middle aged 30-50s and the affect region primarily being left breasts
- FInally I also ran a correlation heat map which gave me insights such as the high correlation between deg-malig to and the recurrence class at a whopping 0.23!
- The heat map also provided some more basic and known observations such as the correlation between age and menopause

What procedure did you use to train the model?

To train the model I:

- Divided the data into train (0.3) and test (0.7) sets
- Used the sklearn for the kNN model
- Used for sklearn Gridsearch for KNN grid search
- Used sklearn SGD for the linear classification
- Checked metrics for each models train and test split

How does the model perform to predict the class?

- For kNN the model assigns a class prediction based on the majority of its k nearest neighbors (class) assignment
- For gridsearch kNN the model runs the same kNN but varies the k values and distance metrics to find the optimal hyperparameters
- For Linear classification the SGD goes through the training data to create a linear classifier, iteratively updating the weights of the classifier with each sample

How confident are you in the model?

Looking at the metrics my confidence varies by model but I'd say I'm most confident about the kNN

- For kNN the overall test accuracy is sufficient at 65%, but the low F1 score of 23.53% shows the model issue in classifying with positive cases
- For the grid search kNN the overall test accuracy is also sufficient at 70.7%, but similarly the low F1 score of 23.26% shows the model issue in classifying with positive cases. Diving deeper into the recall and precision, the model is confident when predicting positives with a precision of (71.4%) but actually still misses many with a recall of (13.89%)
- For linear classification while my test accuracy is the lowest at 61.06%, there is a higher balance between the recall and precision with an F1 score of 29%

Notes on Responsible AI use:

- Used chat GPT to find libraries to assist with data cleaning (drop, duplicates, etc) - but also wrote my own verification code that caught additional data abnormalities
- Used chat GPT for imports and available sklearn libraries
- USed chat GPT for deciphering encoded data