PROJECT 2

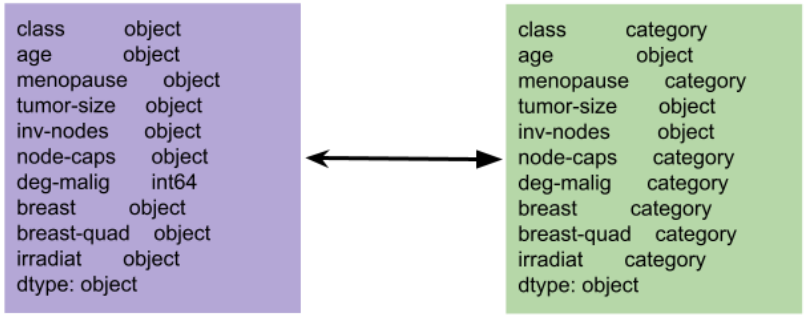
BREAST CANCER RECURRENCE

By: Trey Gower

DATA PREPROCESSING

Type Conversion:

Most the columns in the dataset were of type object, but most the columns were clearly categorical data. This motivated the conversion to type category:



Missing Data:

There were two columns that had missing data which were node-caps and breast-quad. It would be best to train a model (like knn) based on age, tumor-size, and menopause to do some advanced imputation, but I chose to replace the values with the modes of the data instead.

Visualizing:

The univariate plots below give us a good picture of the imbalanced columns in the dataset.

Class: Imbalanced distribution (This is our dependent variable)

Age: Most Cases are within 40-59 year old range

Menopause: Most cases in patients premenopause and menopause cases over age 40

tumor-size: Biggest tumor diameter lies within 30-34, 25-29, and 20-24

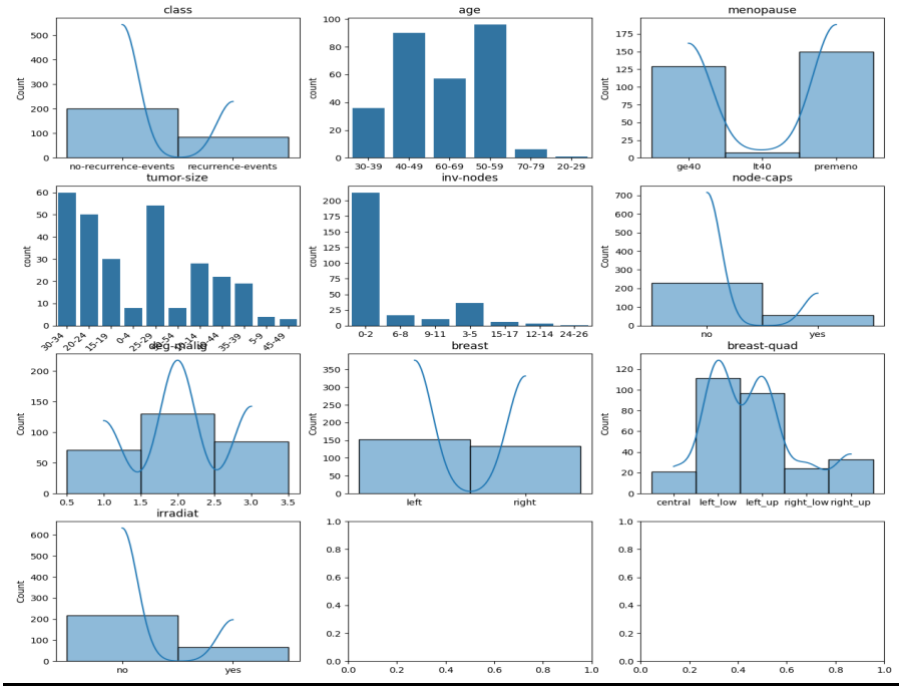
node-caps: Imbalanced distribution

deg-malig: balanced distribution

breast: Fairly balanced distribution

breast-quad: Fairly balanced distribution

irradiat: Imbalanced distribution



One-Hot Encoding:

It was necessary to one hot encode most of the data besides the deg-malig column to train our models.

```
#Define our categorical columns to be one-hot encoded
cat_cols = ['age', 'menopause', 'tumor-size', 'inv-nodes', 'node-caps', 'irradiat']
#Get numeric columns
num_cols = [col for col in combined_data.columns if col not in cat_cols]

#One hot encoding all the categorical variables
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), num_cols),
        ('cat', OneHotEncoder(), cat_cols)
    ])
```

MODEL DECISIONS AND FITTING

Below are the models I chose to use and an honorable mention:

1. **KNN Model**
2. **Logistic Regression Model**
3. **Decision Tree Model**

Honorable Mention:

With the nature of the dataset, it was very imbalanced, it could've been good to use something like Random Forest Classifier to adjust hyperparameters to account for different weighting given a column and its key values. I didn't go this direction as training a single model took quite some time, and I determined the improved accuracy wasn't worth the wall clock time taken.

MODEL METRICS ANALYSIS

Note:

F1-score is the most important here as the data set features imbalanced features (columns), thus accuracy can b e a misleading metric. However, Recall score is very important since we are dealing with determining cancer recurrence and it is part of computing f1-score.

KNN Model Training Data Metrics

Performance on TRAIN
****

	precision	recall	f1-score	support
no-recurrence-events	0.81	0.95	0.88	164
recurrence-events	0.78	0.44	0.56	64
accuracy			0.81	228
macro avg	0.80	0.69	0.72	228
weighted avg	0.80	0.81	0.79	228

Logistic Regression (LR) Model Training Data Metrics

Performance on TRAIN
****

	precision	recall	f1-score	support
no-recurrence-events	0.80	0.93	0.86	164
recurrence-events	0.69	0.39	0.50	64
accuracy			0.78	228
macro avg	0.75	0.66	0.68	228
weighted avg	0.77	0.78	0.76	228

Decision Tree (DT) Model Training Data Metrics

Performance on TRAIN
****

	precision	recall	f1-score	support
no-recurrence-events	0.98	1.00	0.99	164
recurrence-events	1.00	0.94	0.97	64
accuracy			0.98	228
macro avg	0.99	0.97	0.98	228
weighted avg	0.98	0.98	0.98	228

All Models Testing Data metrics:

KNN Metrics:

	precision	recall	f1-score	support
no-recurrence-events	0.66	0.95	0.78	37
recurrence-events	0.60	0.14	0.23	21
accuracy			0.66	58
macro avg	0.63	0.54	0.50	58
weighted avg	0.64	0.66	0.58	58

Logistic Regression Metrics:

	precision	recall	f1-score	support
no-recurrence-events	0.66	0.89	0.76	37
recurrence-events	0.50	0.19	0.28	21
accuracy			0.64	58
macro avg	0.58	0.54	0.52	58
weighted avg	0.60	0.64	0.58	58

Decision Tree Metrics:

	precision	recall	f1-score	support
no-recurrence-events	0.65	0.76	0.70	37
recurrence-events	0.40	0.29	0.33	21
accuracy			0.59	58
macro avg	0.53	0.52	0.52	58
weighted avg	0.56	0.59	0.57	58

How does each model perform to predict the dependent variable?

In comparing the f1 accuracy scores between training and testing data (KNN = .81, .66; LR = .78, .64; DT = .98, .59), we can see that KNN and LR may have been slightly overfitting, while DT was most definitely overfitting our data. Based on this I would throw the decision tree model out. With the two remaining models, KNN and LR, it seems these models performed pretty well in predicting no-recurrence vs recurrence.

Which model would you recommend to be used for this dataset

For this dataset, I would choose KNN. The precision for recurrence events is higher, the recall for recurrence events is slightly lower, but still within an acceptable allowance, and the f1-score for average accuracy is slightly better. All this is without doing a grid search to find the best and smoothest K value to reduce over-fitting.

How does the model perform with respect to false positives and false negatives? Which standard model performance metric is most important to optimize?

In terms of false negatives the KNN model performs very well with no-recurrence at a recall score of .95. However, with recurrence the recall score is only .14, which is very poor and will more times than not report false positives, which is very bad when dealing with cancer readings. Again, to optimize this recall we can do a grid search to find the best KNN K value (I did this in my code).