COE 379L: Project 2

Ashton Cole
AVC687

March 19, 2024

1 Introduction

In this project, we are tasked with building machine learning models to predict breast cancer recurrence. After the source data set is prepared, three models are tested: K-Nearest Neighbor, Naive Bayes, and Random Forest. All are validated against both training and separate testing data.

This assignment demonstrates several important machine learning principles. The data is carefully inspected, cleaned, and analyzed thoroughly before models are even built. A grid search of hyperparameters is used to optimize each model. The models are then evaluated on independent data to determine their worthiness.

2 Methodology

The models learn from a data set of breast cancer patients. Data includes personal metrics, tumor information, and whether or not a recurrence occurred after a certain period of time. Analysis is conducted using a Python 3 kernel in two Jupyter Notebook files. The first file cleans and inspects the data. The second trains and tests the machine learning model.

2.1 Data Preparation

The breast cancer data set initially contains 286 records across 10 columns, defined below in Table 1. First, each column is inspected to see what unique values it holds and whether it has any null or null-equivalent values. Nine records containing a "?" in one of their fields are removed. Categorical variables are converted to a categorical type and represented with one-hot encoding. The `deg-malig` is one of these, becasue although it is an integer, it represents a qualitative scale.

There are a few binned numeric columns: `age`, `tumor-size`, and `inv-nodes`. The machine learning models used accept either binary or continuous variables, so there are two clear options to represent these fields: either as categorical variables with one-hot encoding, or as a numeric value at the center of the bin. Each has its own advantages and disadvantages. One-hot ecoding would make the entire data set binary, allowing methods like the Bernoulli Naive Bayes model to be used, but it also dissolves any ordinality or distancing between bins. For a large number of bins, it also significantly increases the number of columns stored in memory. On the other hand, averaging retains ordinality and distance, but is not fully representative either. It falsely concentrates the data to a discrete set of points on a continuous scale. In this project, the latter option is pursued, although it would be interesting to compare the results against the former.

In addition, some entries are cut from the model as outliers. Plots from the Seaborn library were used to spot such patterns for each variable. Firstly, in the `menopause` column, visualized in Figure 1, only 5 patients report having menopause before the age of 40. This is simply too few data points to inform predictions. Meanwhile, the `inv-nodes` column, visualized in Figure 2, has literal statistical outliers, i.e. values more than three standard deviations from the mean. The vast majority of the set is concentrated in the lower end of the range, making it hard for the model to fit reliably outside of this region. With only 10 records outlying, the safer option is to exclude these too from analysis.

The final data set contains 262 entries and 14 columns, which are described in Table 2.

Table 1: Columns in the raw data set.

Column Name	Type	Meaning
class	object	Whether or not the patient has a recurrence of cancer.
age	object	The age of the patient, organized into 5-year bins.
menopause	object	Whether or not the patient has undergone menopause, and whether that happened before or at and after the age of 40.
tumor-size	object	The size of the tumor in mm, organized into 5 mm bins.
inv-nodes	object	The number of invasive nodes, organized into bins of 3 integers.
node-caps	object	Whether or not the tumor has node capsules.
deg-malig	int64	The degree of malignancy of the tumor, on a qualitative scale.
breast	object	In which breast the tumor resides: left or right.
breast-quad	object	In which "quadrant" of the breast the tumor resides: lower-left, lower-right, upper-left, upper-right, or central.
irradiat	object	Whether or not the initial tumor was treated with radiation therapy.

Table 2: Columns in the processed data set.

Column Name	Type	Meaning
age	int64	The approximate age of the patient, based on the average of the bin.
tumor-size	float64	The approximate size of the tumor in mm, based on the average of the bin.
inv-nodes	int64	The approximate number of invasive nodes, based on the average of the bin.
class_recurrence-events	bool	Whether or not the patient has a recurrence of cancer.
menopause_premeno	bool	Whether or not the patient is premenopausal.
node-caps_yes	bool	Whether or not the tumor has node caps.
deg-malig_2	bool	Whether or not the tumor has a degree of malignancy of 2.
deg-malig_3	bool	Whether or not the tumor has a degree of malignancy of 3.
breast_right	bool	Whether or not the tumor resides in the right breast.
breast-quad_left_low	bool	Whether or not the tumor resides in the lower-left quadrant of the breast.
breast-quad_left_up	bool	Whether or not the tumor resides in the upper-left quadrant of the breast.
breast-quad_right_low	bool	Whether or not the tumor resides in the lower-right quadrant of the breast.
breast-quad_right_up	bool	Whether or not the tumor resides in the upper-right quadrant of the breast.
irradiat_yes	bool	Whether or not the tumor was treated with radiation therapy.

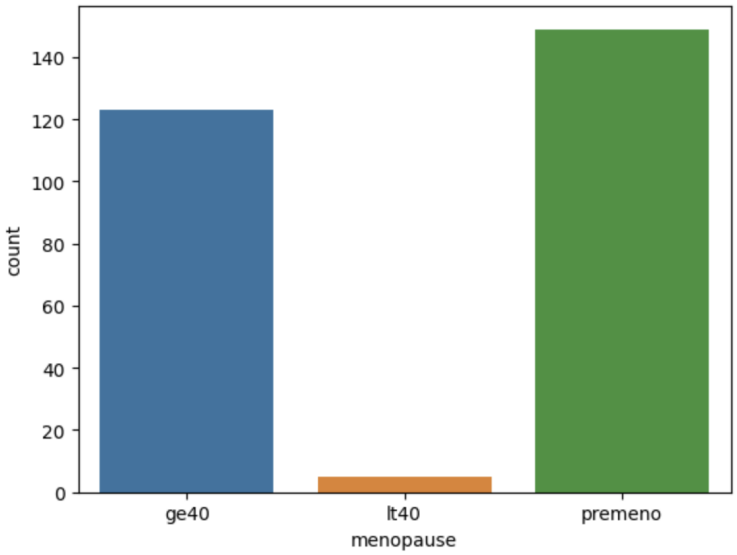

Figure 1: A count plot of menopause.

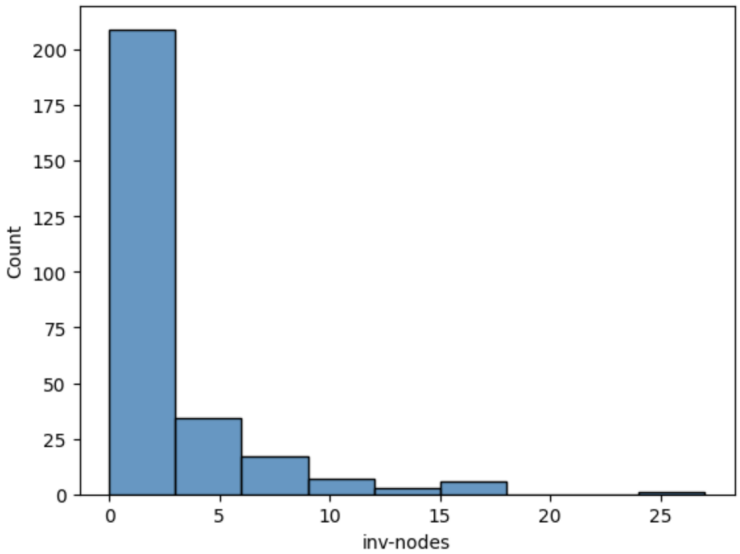

Figure 2: A count plot of inv-nodes.

2.2 Model Training

Before training, the data set is split into two pieces: a training set conisting of a randomly selected 70% section of the data, and a testing set made up of the rest. This ensures that the models can be validated against cases which they have not encountered before, highlighting any overfitting to the training data.

Three models are fit upon the data: K-Nearest Neighbor, Naive Bayes, and Random Forest. These models are sourced from the `sklearn` module. Further, a *grid search* is conducted, where different combinations of model hyperparameters are tested to find optimal models. This search uses cross-validation to further prevent overfitting. Essentially, since the grid search itself needs to test each combination of hyperparameters, it splits the training data into *folds*. In a rotating fashion one of these is used as the test set.

2.3 K-Nearest Neighbor

This model more or less classifies a data point by polling the k closest training data points. The hyperparameters are varied as in Table 3.

Table 3: Hyperparameters for K-Nearest Neighbor Model

Hyperparameter	Values	Description
`n_neighbors`	Range from 1 to 50	The number of nearest points to poll for classification.

2.4 Naive Bayes

Naive Bayse refers to a family of methods, including Gaussian, Multinomial, and Bernoulli. In this case, specifically the Multinomial Naive Bayes method is used. This is the model that works best for classifying discrete features, appropriate for a data set of booleans and binned scales. The Gaussian method is not appropriate, because it relies on the asssumption that all input variables are normally distributed. The Bernoulli method, however, would be appropriate if the binned scales were represented with one-hot encoding, since it requires all inputs to be binary. No hyperparameters were considered for this method.

2.5 Random Forest

A Random Forest is based on Decision Trees. This machine learning method classifies data points using sequential boolean expressions arranged into a binary tree. The "leaf" nodes are where predictions are made. It tends to over-fit the data, so a Random Forest combines the results of several decision trees to make a decision. The hyperparameters are varied as in Table 4.

Table 4: Hyperparameters for Random Forest Model

Hyperparameter	Values	Description
`n_estimators`	Range from 10 to 100 by 2's	The number of "trees" in the "forest."
`max_depth`	Range from 2 to 20	The maximum number of layers permitted in any tree.
`min_samples_leaf`	Range from 1 to 5	The minimum number of samples permitted to justify a leaf node on a tree.
`class_weight`	0: 0.3, 1: 0.7 0: 0.5, 1: 0.5 0: 0.7, 1: 0.3	A weighting associated with the output parameter, applied to creating brances in new trees.

Table 5: Performance metrics for all machine learning models.

Model	K-Nearest Neighbor		Multinomial Naive Bayes		Random Forest	
Data	Training	Testing	Training	Testing	Training	Testing
Accuracy	0.754	0.633	0.727	0.747	0.781	0.679
Recall	0.321	0.136	0.453	0.500	0.642	0.455
Precision	0.654	0.231	0.533	0.550	0.618	0.417
F1 Score	0.430	0.171	0.490	0.524	0.629	0.435

3 Results

Each model is validated on both the original training data and new testing data. The predictions for `class` against the original data are evaluated using accuracy, recall, precision, and f1 scores. Essentially, accuracy is the proportion of correct predictions, recall only penalizes for false negatives, and precision only penalizes for false positives. The f1 score is the harmonic mean of recall and precision. A summary of these metrics is shown in Table 5.

4 Analysis

In deciding on the best model, we need to cosider not only the raw metrics, but a broader context of the implications of making false predictions. There are two types of errors that can be made: a false positive and a false negative. For this data set, a false positive would be predicting recurrence when no recurrence occurs. Meanwhile, a false negative would be predicting full remission when a tumor does come back. Were this model used as an aid in a diagnostic context, the latter, a false negative, would be far worse. Additional screening and imaging quickly and inexpensively rule out a false positive. On the other hand, a false negative would offer a false sense of security allowing a tumor to grow and metastasize unchecked, endangering the life of the patient. Thus, it is important to pick a model which minimizes false negatives.

What this means is that additional weight should be given to the model having a high recall, i.e. few false negatives. Weight should also be given to the results for the testing sets, as the models could be overfit to the training data.

Considering these factors, the Multinomial Naive Bayes model seems to perform best, followed by Random Forest and K-Nearest Neighbor. It has high scores in all categories, performs best of the three on testing accuracy and recall, and shows the least overfitting. Ironically, it actually scores slightly higher for the independent testing data in all scoring categories.

5 Conclusions

The Multinomial Naive Bayes model seems to be the all-around best model for predicting breast cancer recurrence. It is hardly a perfect model, with its approximately 75% accuracy preventing it from ever being used as a tool for diagnosis. Even as a risk-identifier, it risks causing false alarm or a false sense of security in a large number of cases.

Perhaps there are better models that could predict the data set, but given that several different models show relatively close scores, we should entertain the idea that the issue lies in the data itself. Cancer is unpredictable. After treatment and surgery, remission is a probability, not a guarantee. As an aside, when my mother had liver cancer in 2010, she had annual follow up appointments for severeal years following her transplant. After almost 15 years, the chances that the same original cancer will return is essentially none, but in the time in between, there was a continuously decreasing probabability of recurrence. Factors like age or tumor characteristics might hint at a tumor's hardiness. However, it is important to remember that the physical mechanism of recurrence, the retention of cancerous cells in the body which grow into new tumors, is a microscopic phenomenon which is difficult to measure, and hardly capturable in the columns provided. More expansive data from deeper measurements is likely the key to building a better model.