Cancer Recurrence Prediction Machine Learning Models

Jackson Thetford
COE 379L
Software Design for Responsible Intelligent Systems

Introduction

The purpose of this project is to apply exploratory data analysis skills to process data and apply it to train various machine learning classification models. The data given for this project is a breast cancer dataset containing features such as age, degree of malignancy, tumor size, etc., and if recurrence events have occurred. This project focuses on the processing and visualization of the data and then using the processed data to build 3 machine learning classification models to predict if a patient will have recurrent breast cancer. The classification models that were chosen for this project were the Random Forest Classifier, the K-Nearest Neighbor Classifier, and the Logistic Regression Classification model. Analysis of the statistical metrics accuracy, recall, precision, and f1-score are tested and used to validate the models.

Data Preparation

The dataset for this project is a breast cancer dataset which consists of 286 entries with 10 variables associated with breast cancer: class (recurrence or no recurrence of breast cancer), age, menopause stage, tumor size, invasive nodes count, node caps count, degree of malignancy, breast (left/right), breast quadrant, and irradiation (yes/no). The data set contains all ‘object’ data types, except for the degree of malignancy attribute which is of data type integer.

To prepare this data for modeling the following steps were done:

1. **Dropping Duplicate Entries:** It was found that there were 14 duplicated entries. These were dropped.

2. **Filling Missing Values:** There are missing values in the 'node-caps' (8) and 'breast-quad' (1) attributes. As the dataset consists of only 286 entries, it was determined that filling the values rather than deleting the row would be best as to not lose valuable data. The missing values were filled with the mode for their respective attribute.

3. **Ordinal Encoding:** The "age", "tumor-size", and "inv-nodes" variables are all ordinal variables that contain an inherent hierarchy or ranking in their values. These variables are also given as ranges (bins) rather than specific values. One-hot encoding would have destroyed the relationship/ranking of these variables, so these variables were ordinally encoded.

4. **One-hot Encoding:** The "class", "menopause", "node-caps", "breast", "breast-quad" and "irradiate" variables were one hot encoded as they are purely categorical, or nominal variables. This split the columns up into sub columns such as 'class_recurrence-events', ‘menopause_lt40’, and ‘menopause_premeno’ that are all numeric, Boolean columns that the upcoming models can interpret.

The data was analyzed and visualized, which is shown in the Jupyter Notebook. The frequency plots for each attribute are shown in the Jupyter notebook.

Model Training

To train the linear regression model the scikit-learn library was used. The objective of the model was to predict if a patient will have recurrent breast cancer using the 9 independent variables in the dataset. The procedure used train the model was as follows:

1. **Data Splitting:** First, the target variable "class_recurrence-events" column was isolated and saved as the independent variable "y". The other 8 independent variables were saved under the variable "X", most of which appeared as one-hot or ordinally encoded columns. The X variable DataFrame excludes the "class_recurrence-events" variable. This data (the independent and dependent variables) was then split into training and testing sets using scikit-learn's `train_test_split` function, with 70% of the data allocated to the training set and the other 30% to the testing set. Stratification sampling was used to split the data into testing and training sets that preserves the same proportion of each class of dependent variable.

2. **Parameter Grid:** Several arrays of hyperparameters for each model were converted to a parameter grid representing all the possible combinations of given parameters to be tested in making different models to find the best model. This process allows for the tuning of hyperparameters in order to find the optimal values for each model.

3. **Model Training & Grid Search Cross Validation :** The training data was then given to several classification models: Random Forest Classifier, the K-Nearest Neighbor Classifier, and Logistic Regression. For each model, a Grid Search Cross Validation object was created from the 'sklearn.model_selection' module and the fit method was used to test all combinations of the hyperparameters in the parameter grid, as mentioned above, on the training data. A 5-fold cross validation was used with the scoring metric being recall.

Model Analysis

To evaluate the model's performance, accuracy, recall, precision, and f1-score were all used, however there was a greater weighted interest in the recall metric for each model. In the context of predicting breast cancer recurrence, the number of false negatives should be minimized. It was decided that the model would be much more applicable and useful if it was more accurate in predicting true negatives (minimizing false negatives) as the cost of a false negative is much more serious than incorrectly identifying a negative case as positive. In other words, it would be better to tell a patient they have recurrent breast cancer, when they don't, as the only harm that would come from that is most likely more frequent tests. On the other hand, telling a patient with recurrent breast cancer that they don't have recurrent breast cancer would lead to more serious, potentially life-threatening, outcomes.

Analyzing the models, the Random Forest Classifier's performance can be summarized by the following metrics, where the performance on the test data is most important:

	Accuracy	Recall	Precision	F1-score
Test Data	0.29	1.00	0.29	0.45
Train Data	0.31	1.00	0.30	0.46

Random Forest Classifier Metrics

The K-Nearest Neighbor model performed poorly with the following metrics:

	Accuracy	Recall	Precision	F1-score
Test Data	0.74	0.42	0.59	0.49
Train Data	0.79	0.40	0.79	0.53

K-Nearest Neighbor Classifier Metrics

The Logistic Regression model preformed slightly worse with the following metrics:

	Accuracy	Recall	Precision	F1-score
Test Data	0.72	0.38	0.53	0.44
Train Data	0.72	0.23	0.59	0.33

Logistic Regression Classifier Metrics

Overall, the Random Forest Classifier shows exceptional performance in terms of recall, achieving a perfect score of 1.00 on both test and train data, indicating it correctly identifies all true positives. However, its accuracy and precision are significantly lower at 0.29 for the test data, meaning a high rate of false positives. The F1-score, which balances precision and recall, is relatively low at 0.45 for the test data, indicating an imbalance between precision and recall.

The K-Nearest Neighbor (KNN) Classifier has a more balanced performance across metrics but with a lower recall of 0.42 on test data meaning it missed a significant number of true positives. Its accuracy on test data is 0.74, and precision is 0.59, showing a better balance but at the cost of lower recall.

The Logistic Regression model presents the lowest recall of 0.38 on test data among the three, suggesting it fails to identify a considerable number of actual positive cases. Its accuracy and precision on test data are slightly better than the Random Forest model but still not optimal, with values of 0.72 and 0.53, respectively.

Overall, the Random Forest Classifier is the most recommended model to predict recurrence in breast cancer. It does not perform well in accuracy or precision, but it is consistently perfect in recall predicts an extremely low number of false negatives. As the goal of these models is to predict the recurrence of breast cancer, the models should be scored and incentivized to minimize the possibility of false negatives, or in this case, classifying someone as having non-recurrent breast cancer when they have recurrent breast cancer. This means there should be emphasis on the recall scoring metric of the model.

Resources

[1] Data: https://raw.githubusercontent.com/joestubbs/coe379L-sp24/master/datasets/unit02/project2.data

[2] Class Repo: https://coe-379l-sp24.readthedocs.io/en/latest/index.html

[3] ChatGPT: https://chat.openai.com
(Used for visualization formatting / generating some figures)