COE 379L Project 1 Report

Introduction

This project focuses on analyzing the recurrence of breast cancer in patients through the "Breast Cancer Dataset" published on the UCI Machine Learning Repository by the Oncology Institute. In the following report, I will explain how I prepared the provided data for model training, some insights I gained from this data preparation, what procedures I used to train this model, how well the model performs in predicting the class, and how confident I am in the application of this model for breast cancer recurrence prediction.

Note on ChatGPT Use

For this project, I used ChatGPT to help explain what the several features of the dataset mean and what each of their values mean. I used it to help me figure out how to map unique values of a pandas series to numerical values (.map function), generate the custom_sort function I used for displaying the DataFrame columns in count plots for univariate analysis, and to debug an issue I had with displaying classification results for the SGD classifier where I used "zero_division=0" to suppress some warnings. Additionally, I used it to combine the notebook cells from cancer_eda.ipynb and cancer_classification.ipynb into breast_cancer.ipynb since the assignment required one notebook for submission for Parts 1 and 2.

What did you do to prepare the data?

First, the data provided had 10 columns total, 9 of which were feature independent variables and the final column, class, was the target dependent variable. I performed some basic exploratory data analysis on the dataset, examining the shape and size of the data, which has 386 rows with 10 columns. Next, I found that there were 11 duplicate rows in the dataset, which I removed, and I noticed that the tumor-size and inv-nodes columns had null values. I also found that node-caps and breast-quad had unknown values of '?' and '*' which needed to be replaced.

In order to do this, I used the groupby and transform method mentioned in lectures earlier to group each column by the two most relevant features to them and replace missing values with the mode of the values in their groups. After this, I re-named class to is_recurring, node-caps to lymph_node_capsular_invasion, breast to is_left_breast, and irradiat to taken_radiation_therapy for clarity on which column means. Likewise, I converted these columns from objects to integer values by mapping each of their two outcomes to 0 and 1 values.

Next, I visualized the dataset through basic univariate analyses. I computed the basic statistics of integer columns in the dataset that included the count of rows, the mean, standard deviation, min, max, and the 25/50/75% values of each column. Additionally, I visualized all columns through count plots that show the range of values and the frequency of each value.

Finally, I performed one-hot encoding on the remaining object columns to convert them all to boolean columns, and then to numerical data that the classification model can process easily (0/1 instead of True/False). I saved this cleaned dataset to a new file for later processing.

What insights did you get from your data preparation?

From the dataset statistics mentioned earlier, I found that for each of columns, the count of rows is 375. The mean of deg-malig is 2.07, indicating that most of the tumors are moderately malignant. The mean of lymph_node_capsular_invasion, taken_radiation_therapy, and is_recurring are all within the range of 0.2 to 0.3, indicating that the presence of node caps, the need for irradiation, and the recurrence of breast cancer are all relatively rare. The mean of is_left_breast is roughly 0.5, indicating that the dataset has roughly equal numbers of left and right breast cancer patients.

From the count plots shown in the Jupyter notebook, we can see that the dataset has more middle-aged patients from 40 to 60 years old. Most of the patients are in premenopause and ge40 menopause (post-menopause), but only a few are in the lt40 group. The tumor sizes are most frequent in the 25-34 mm range, followed by the 20-24 mm range. The number of involved nodes is most frequent in the 0-2 range, followed by the 3-5 range. The presence of node caps, the number of people who have had radiation therapy, and the number of people who have had breast cancer recurrence are all relatively rare, while the location of left or right breast is evenly split. The degree of malignancy is most frequently 2 (medium), followed by 3. Finally, the breast quadrant is most frequent in the left lower quadrant, followed by the left upper quadrant.

Additionally, one thing to note is that for this problem, recall is the most important metric. This is because we want to minimize the number of false negatives. False negatives in this case would be a recurring breast cancer case that was predicted to not recur. This would be a very serious mistake as the patient would not receive the necessary treatment to prevent the recurrence of breast cancer. Therefore, we want to minimize the number of false negatives and maximize the number of true positives. This is why recall is the most important metric for this problem, and why we use recall as the metric for scoring the classification models during training.

What procedure did you use to train the model?

To train classification models on this dataset, I first split the data into training and test splits using the train_test_split method from scikit-learn, with 70% as training and 30% as test, and stratifying across the y (is_recurring) column. After this, I trained 3 models on the data: K-Nearest Neighbor (KNN) Classifier, KNN Classifier with GridSearchCV (GSCV), and Linear SGD Classification. For the first KNN model, I arbitrarily chose a n_neighbors value of 20 and fit the model on the data. For the KNN GSCV model, I set up a hyperparameter search space of n_neighbors from 1 to 100, cross validation size of 5 folds for each training, and scoring function as the recall formula. From this, I found surprisingly that n_neighbors = 1 had the best performance. Lastly, I fit the SGD classifier with alpha = 0.01 and perceptron loss on the data.

How does the model perform to predict the class?

The regular KNN Classifier has an average accuracy of 0.7, a precision of 0.75 for label 1 (is recurring) and 0.7 for label 0 (is not recurring), a recall of 0.08 for label 1 and 0.99 for label 0, and an F1 score of 0.15 for label 1 and 0.82 for label 0, for the testing set. For the KNN GSCV, it has accuracy of 0.57, precision scores of 0.69/0.33, recall of 0.66/0.36, and F1 score of 0.68/0.35 for labels 0/1 on testing set. Lastly, for SGD classification, it has accuracy of 0.68, precision scores of 0.68/0.5, recall of 0.99/0.03, and F1 score of 0.81/0.05 for labels 0/1.

How confident are you in the model?

Based on the results from above, the regular KNN classifier is good at predicting non-recurring cases (high true negatives, low false positives), but bad at predicting recurring cases (many false negatives). The KNN GSCV model is more balanced but still struggles with recurring cases. It’s better than regular KNN in recall for label 1, but lower precision for both labels. Lastly, the SGD classifier is similar to regular KNN in that it’s good at predicting non-recurring cases and very poor at predicting recurring cases. All of these models struggle with prediction of recurring cases, and so while you can be fairly confident that recurrence is unlikely when the model outputs a negative, you can’t trust the model when it outputs recurrence as positive since its scoring metrics for those are so low.

One thing that could be hindering the performance of these models is the imbalance of classes in the dataset as there are 254 non-recurring cases while there are only 121 recurring cases, less than half as much. In order to mitigate this, the models would have to take into account the weights of each class when training so that they can be more accurate in their predictions. This would be a next good step to improve upon the existing models and to evaluate if this change would bring about any increase in recall, F1, or accuracy scores.