Project 2

COE 379L

Joe Stubbs

Braulio Lopez

In Project 2 we were tasked to load a dataframe and work around it by using data analysis and machine learning skills. This project aims to build a machine learning model using supervised learning techniques that can accucrately predict how many patients will have recurrence of breast cancer. The dataset used for this project is a Breast Cancer dataset which contains 10 different columns that provide information of a group of patients with the disease. The data frame contains the following columns: Class (No-recurrence and recurrence events), age (binned age for sample population), menopause (menopausal status of patient), tumor-size (size of tumor binned in mm) Inv-nodes (invasive nodes binned), node-caps (Node capsule), deg-malig (degree of malignancy), breast (left or right), breast-quad (left-up, left-low, right-up, right-low, central) and irradiat (irradiation).

Before training our model, the dataframe needed to be prepared so it can make a good model of what we want to predict. To prepare the data it was necessary to load it using the pandas library. After loading the dataset it was crucial to get information of the columns, in particular to check if the variables presented needed data type conversions. In this case the dataframe does not need datatype conversions as all the other variables are object type which is good for categorical variables, the only int datatype was the degree of malignancy which is correct as it is a whole number. After checking if the columns needed datatype conversions, checking if data is missing is a crucial step. Based on the analysis, there were nine missing values, eight from the node

capsule column and one from the breast quad location. To make our model more accurate it was important to not drop these rows that contained the missing values instead the mode of column were replaced on these missing values to make accurate predictions. The data was also prepared with one-hot encoding algorithms that converts the categorical variables into a form that can be implemented to machine learning model. To do that, all the columns except the degree of malignancy column, were converted as category type and after that we get the dummies using the get_dummis function that creates binary columns for this categorical variables.

The techniques used to train the model were K-nearest neighbor classifier, decision tree and logistic regression. The performance for all models was presented on the accuracy, recall, precision and f1-score tables which were printed on the jupyter notebook. For the K-nearest neighbor classifier, the model used the hyperparameter tuning using GridSearchCV() which indicated an accuracy of 69% on the test data and 74 % on the train data. On the other hand, looking at the Decision tree model we can see that it successfully managed to make a decision tree of predicting models. It showed an accuracy of 65% on the test set while also indicating an accuracy of 97% on the train set. Comparing this to the logistic regression model, the accuracy of the test data increased to 72 % and a 76% on the train set.

From all the models discussed I recommend to use the model of logistic regression as it shows a more accurate prediction on the model. It contains a more balance accuracy from the test and training sets compared to the K-neearest neigbor classifier. The logistic regression also provided a good tradeoff between precision and recall and this is something really important in the medical field.

The performance metric that is essential to optimize is “recall” as the cost of missing a false negative can be higher than missing a false positive which can lead to the patient not getting treatment right away and their condition to get worse. Therefore it is important to optimize this metric while maintaining high accuracy and precision to make our model more exact.

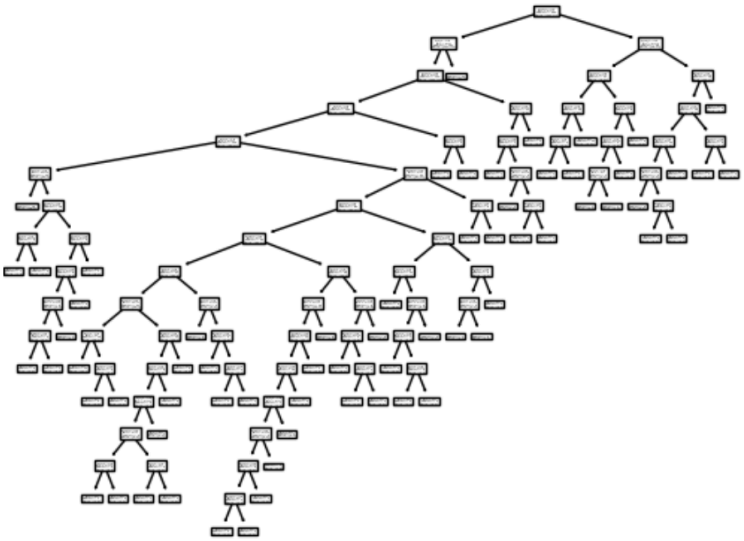
Results:

K- nearest neighbor classifier:

Accuracy of knn on test data is : 0.69
Accuracy of knn on train data is : 0.86
Accuracy on the test data set for the model produced with the optimal k is: 0.6976744186046512
Accuracy of on train data for the model produced with the optimal k is: 0.74

Decision Tree:

```
Performance on TEST
********************
              precision    recall  f1-score   support

       False       0.72      0.82      0.77        60
        True       0.39      0.27      0.32        26

    accuracy                           0.65        86
   macro avg       0.55      0.54      0.54        86
weighted avg       0.62      0.65      0.63        86

Performance on TRAIN
********************
              precision    recall  f1-score   support

       False       0.96      1.00      0.98       141
        True       1.00      0.90      0.95        59

    accuracy                           0.97       200
   macro avg       0.98      0.95      0.96       200
weighted avg       0.97      0.97      0.97       200
```



Logistic Regression:

```
Performance on TEST
********************
              precision    recall  f1-score   support

       False       0.75      0.90      0.82        60
        True       0.57      0.31      0.40        26

    accuracy                           0.72        86
   macro avg       0.66      0.60      0.61        86
weighted avg       0.70      0.72      0.69        86

Performance on TRAIN
********************
              precision    recall  f1-score   support

       False       0.78      0.92      0.84       141
        True       0.67      0.37      0.48        59

    accuracy                           0.76       200
   macro avg       0.72      0.65      0.66       200
weighted avg       0.75      0.76      0.74       200
```