Software Design for Intelligent Systems Project 2

Serena Shah | ss94574 | serena.shah@utexas.edu

The dataset being analyzed was one that provided technical information about over 280 breast cancer patients. This information included patients' age ranges, tumor location, tumor size, and menopausal status. To prepare the data for analysis, it was preprocessed in a few ways.

First, the data in `csv` format was read into a `pandas` dataframe. After parsing through the data, it was revealed that two columns had invalid values of `'?'` for some rows: the `node-caps` column and the `breast-quad` column. The `node-caps` column values should've been either `'yes'` or `'no'`; the `breast-quad` column values should've been either `'left_low'`, `'right_up'`, `'left_up'`, `'right_low'` or `'central'`. These rows' invalid values were substituted with the mode of that column, as mean and median values are strictly numeric statistics and cannot be applied to categorical data.

Next, the data was visualized via univariate analyses plots to better understand relationships between variables. These analyses revealed there are a higher number of breast cancer patients in the `premeno` and `ge40` menopausal categories. Additionally, tumors are more frequent in the left breast, though not significantly. The last analysis revealed that the spread of tumor sizes for breast cancer patients is largely normal, though slightly skewed right, with the majority of tumors being between 20 and 34 mm.

Finally, the columns with categorical data were converted from type `object` to type `category`, which were all variables but `deg-malig` of `int` type. All categorical variables were then one-hot encoded into a binary vector in order to improve performance and efficiency during machine learning implementation. These columns with *n* categories were split into *n-1* columns of `bool` type during one-hot encoding.

Three models were used to implement a machine learning prediction of patients with recurrences of breast cancer. The recurrence status was predicted using three supervised learning models supported by the `sci-kit learn` library: K-Nearest Neighbor Classifier, Multinomial Naive Bayes, and Decision Trees. All three models were trained using a 70:30 training to testing reproducible split. For the K-Nearest Neighbor classification, the *k* hyperparameter was optimized for search accuracy in a cross-validation with 5 folds. The optimal *k* value was found to be 7. For Naive Bayes, the multinomial subtype was chosen due to the discrete nature of the feature variables and its compatibility with categorical data. The decision tree did not need any hyperparameter or specification with subtype. The table below shows a summary of the classification scores/performance metrics of each model.

	Precision	**Recall**	**F1**	**Accuracy**
KNN (k=7)	0.75	0.36	0.49	0.78
Multinomial Naive Bayes	0.31	0.92	0.46	0.37

Decision Tree	0.45	0.52	0.48	0.67

The most accurate model is by far the KNN model which was optimized for accuracy and received an accuracy score of 0.78. Multinomial Naive Bayes has very low accuracy of 0.37, and the Decision Tree's accuracy is fair at 0.67. Based on this information, it looks as though the most accurate predictor in a patient having a recurring breast cancer tumor is the KNN model. However, it is important to note that the recall of the KNN model is significantly low, which is problematic considering the intentions of the model prediction. Because the machine learning model is aiming to predict recurrence of breast cancer in patients, it is more important for it to minimize the presence of false negatives, in which a patient who has a recurrence of breast cancer is wrongly predicted to not have one and misses potentially life-saving treatment. Because false negatives are looking to be minimized, our model should be aiming for an optimal recall, in which false negatives are lower.

Fortunately, the KNN model can be optimized for different model qualities. The KNN model was optimized to search for recall when choosing the *k* hyperparameter. This yielded a *k* value of 1, and the statistics below on scoring.

	Precision	**Recall**	**F1**	**Accuracy**
KNN (k=1)	0.56	0.56	0.56	0.74

The recall significantly improved and the accuracy dropped just by 0.02, a good trade-off for the factor we want to minimize (false negatives). Compared to the

other models, KNN (k=1) preserves accuracy in the model and has significantly improved recall, the most important metric to observe. Perhaps the Decision Tree would be the better option with the most optimal recall of 0.92, though it seems to have been overfit to the training data, largely reducing accuracy.