Machine Learning for Breast Cancer Diagnosis

Capstone Project – Machine Learning Engineer Nanodegree

Author: Harini Sridhar

I. Definition

Project Overview: Breast cancer, one of the leading causes of cancer deaths in women today, is a condition in which malignant (cancer) cells form in the tissues of the breast. In 2018, an estimated 266,120 new cases of invasive breast cancer are expected to be diagnosed in women in the U.S and about 40,920 women are expected to die [1]. The death rates have been dropping steadily since 1989, as a result of treatment advances, earlier detection through screening, and increased awareness.

The most common breast cancer screening test is a mammogram, which is an x-ray of the breast. The ability of a mammogram to find breast cancer may depend on the size of the tumor, the density of the breast tissue, and the skill of the radiologist. However, scanning through mammograms for diagnosis is time-consuming and challenging even for expert radiologists. Findings on a mammogram leading to further recall are identified in approximately 5%-10% of patients, even though breast cancer is ultimately confirmed in only three to ten cases in every 1000 women screened. To overcome the known limitations of human observers, additional reading of mammograms by another radiologist are often obtained before making the final diagnosis. However, this approach is neither scalable nor financially viable. Computer technology is proving to be helpful in this regard. One very promising adaptation is computer-aided detection (CAD) in mammography [2]. A CAD system is usually comprised of the following stages:

- *Candidate generation* – identifies suspicious unhealthy candidate regions of interest (ROI) from an x-ray image;
- *Feature extraction* – computes descriptive features for each candidate so that they can be represented by a vector of numerical values or attributes;
- *Classification* – differentiates candidates that are malignant cancers from the rest of the candidates based on the extracted features; and
- *Visualization* – presentation of CAD findings to the radiologist.

Problem Statement: While each of the four stages in a CAD system is open-ended and warrants effective solution design, the goal of this project is to utilize machine learning for the *classification* stage – predict whether a *candidate* tumor region from the x-ray is malignant or benign. For this purpose, we use a dataset that consists of x-ray images from patients from both categories of tumor, i.e. posed as a binary classification task. Note that, the ROI definition and feature extraction stages were carried out *a priori* using standard tools, and this project will build a predictive modeling solution for diagnosis based on the features.

Datasets and Inputs: The dataset used for this project was curated as part of the KDD Cup competition organized in 2008 [3] and was originally provided by Siemens Medical Solutions. A breast cancer screen typically consists of 4 X-ray images; 2 images of each breast from different directions giving a total of 4 images per patient. Each image is represented by several candidates. Each candidate has 117 features, and a class label indicating whether or not it is malignant. The dataset consists of a total of 102,294 candidates corresponding to 1712 patients, but only an extremely small fraction of these candidates (~650) is actually malignant. Following common practice in machine learning, this dataset will be split into training and validation sets for analysis and benchmarking. The dataset can be downloaded from [4].

Solution Statement: The data will first be preprocessed and split into train/test sets. Different classifier models will be fit, along with hyper-parameter tuning, to evaluate their effectiveness on imbalanced datasets. In order to achieve variance reduction, we will design an ensemble classifier, wherein the best performing classifiers from the previous step will be used as the base model. Resampling techniques will be used to deal with the dataset imbalance and classifier models will be rebuilt to understand their robustness. Finally, we will build a *Super Ensemble* by designing a strategy to combine different sophisticated models from the previous steps. The designed models will be evaluated using metrics such as precision/recall, specificity, F1 score, Geometric mean, and response receiver operating curves (ROC).

Metrics

Evaluating the quality of the designed machine learning models is particularly challenging due the high imbalance in the label distribution. Hence, we adopt a number of popular metrics for measuring model fidelity in order to obtain a holistic evaluation. The most common metric in classification tasks, overall accuracy is not suitable for imbalanced datasets, since it is overly optimistic. For example, the overall accuracy can be over 99% without detecting even a single malignant case. Hence, this metric will not be used in this project.

- Precision and Recall: The precision and recall of individual classes can be chosen as a metric. In medical diagnosis scenarios, false negatives are more severe than false positives. Hence, we want to achieve high recalls.
- Specificity: This indicates the true negative rate, i.e., ability to correctly identify benign tumors.
- F1-score: This is the harmonic mean of Precision and Recall. It indicates the balance between the precision and recall. Its value can be between 0 and 1, 1 being the best value.

$$F1\ score = \ precision * recall \ / \ (precision + recall)$$

- Geometric Mean: This metric combines true negative rate and true positive rate at a specific threshold - where both the errors are considered equal.
- ROC curves: An ROC curve indicates the true positive rate (sensitivity) as a function of the false positive rate (1-specificity). FROC is similar to ROC, except that the false positive rate on the x-axis is replaced by the number of false positives per image. Finally, we measure the area under the curve.

II. Analysis

Data Exploration

The dataset provided has two files - the “Features.txt” file contains the 117 input features for 102294 samples. The “Info.txt” file contains 11 columns for the same number of samples. These columns correspond to the target label (benign or malignant), patient ID, lesion ID, and (x,y) coordinates of the ROIs. The set of coordinate values form 7 columns of the “Info.txt” and are added to the input features, thus forming a total of 124 input features.

From doing an initial analysis of the dataset, the following distributions are seen in the ROIs and patients –

Number of Benign cases = 101671
Number of Malignant cases = 623

Total Number of patients = 1712
Number of Benign patients = 1594
Number of Malignant patients = 118

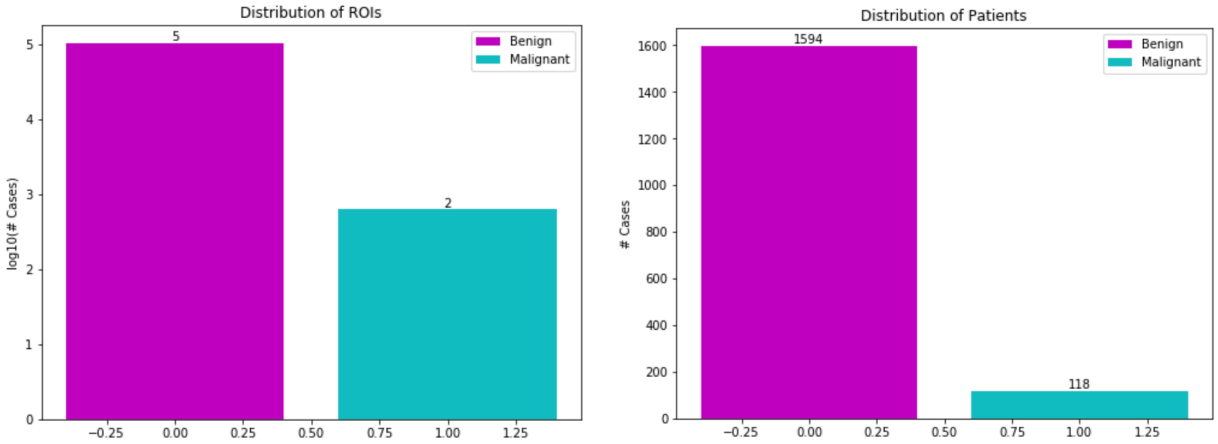

Figure 1. Basic statistics from the dataset. The label distribution is highly imbalanced, and majority of the population belongs to the malignant class.

Figure 1 shows the label distribution for this dataset. This dataset presents crucial challenges that need to be considered during the solution design: First, the dataset has an imbalanced class distribution. Only a very small fraction of the dataset is actually malignant. Most popular classification methods are effective only when the datasets are balanced. Consequently, it is important to understand their behavior in this constrained scenario. Second, the large number of features in the data can make model design challenging, in terms of generalizability. Third, the dataset has some inherent challenges (multiple scans for the same patient), thus requiring extensive pre-processing. The project aims to overcome these challenges and build a good predictive model.

Exploratory Visualization

PCA to visualize the entire dataset: Using Principal Component Analysis (PCA), one can obtain 2-D embeddings for visualizing high-dimensional data. From the 2-D scatterplot in Figure 2, one can observe that the malignant and benign classes are not easily separable and hence simple linear classifiers cannot be very effective. This is inferred from the observation that there are magenta samples (benign) behind the blue samples (malignant). But there is lot of variability in the benign class, although they are not confused with malignant class. This characteristic, along with the high imbalance in number of malignant candidates, can easily make a classifier overly optimistic by flagging every candidate to be benign.

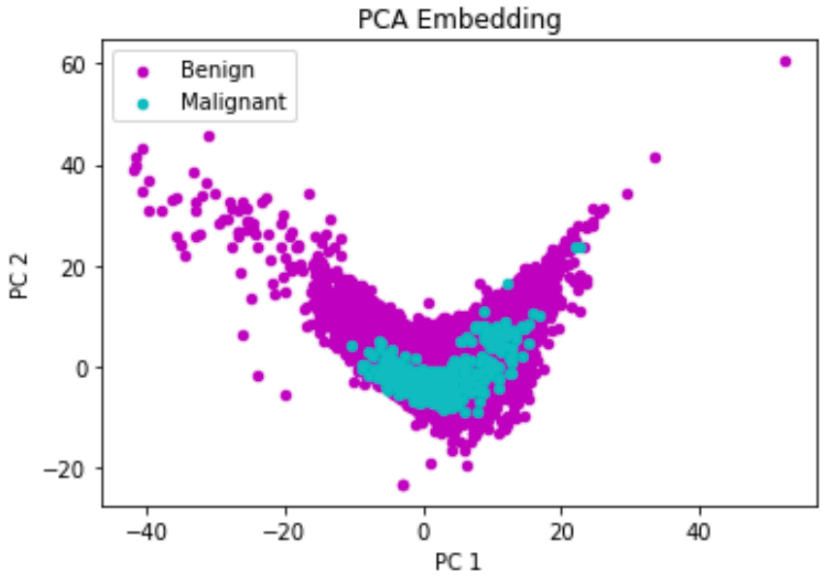

Figure 2. Visualizing the dataset using PCA embeddings of the high-dimensional feature vectors. It can be observed that the two classes are not easily separable.

Algorithms and Techniques

Given the inherent challenges with this dataset, it is not straightforward to determine which machine learning technique would be suitable. We will explore a number of algorithmic choices and perform rigorous evaluation. Insights from such an analysis will provide insights on applicability of different techniques on real-world problems such as this. More specifically, this project the use of different ensemble learning techniques, resampling strategies and a parameterized combination of multiple effective models (referred as a *super* ensemble). These approaches are briefly described below. We use simple classifier models (Decision Trees, Linear SVM and Naïve Bayes) that are usually effective on balanced datasets as benchmarks, in order to illustrate the difficulty of this problem.

Ensemble Methods: In order reduce the variance of complex classifiers, i.e., avoid overfitting, ensemble methods aggregate predictions from multiple (often weaker) models trained to be effective at different aspects of data. While *Bagging* methods fit multiple models using randomized bootstraps, Boosting approaches progressively build constituent models to refine the overall

performance by focusing on regions of data domain where the previously trained models fail. In particular, we will train (i) Bagging classifier on the best performing benchmark method, (ii) Random Forest classifier – Bagging with decision trees, (iii) AdaBoost classifier, (iv) Gradient Boosting classifier. Note that, unlike bagging approaches, boosting can reduce variance and bias simultaneously, thus producing powerful models. However, under data imbalance, their behavior is not obvious. Hence, ensemble methods form first set of techniques that we propose to employ.

Resampling for Dealing with Imbalanced Data: While changing the classification technique provides different relaxed assumptions on data or enables approximation of complex classification surfaces, the inherent problem with the data persists. Hence, in addition to choosing the right technique, it is equally critical to attempt to fix issues with the data. In our case, class imbalance is the key issue and this is generally prevalent in many applications in the medical domain. We propose to use resampling techniques, which can be used to either augment datasets or subsample datasets to reduce imbalance, such that the models are more robust. Data augmentation is a widely-adopted strategy to make balanced dataset through creation of new samples by applying meaningful geometric transformations to the original data. For example, given a scan with tumor, one can arbitrarily translate the tumor to different locations of the organ to expand the dataset with fictitious patients. However, in this problem, we do not have access to the original images, and there is no sufficient information about the features to apply such transformations. Hence, we use resampling to subsample datasets. Broadly, there are two kinds of sampling that can be used – (a) undersampling the negative samples since they are overwhelmingly common, compared to positive (malignant) cases – clustered resampling is a popular example, (b) undersampling the negative class while oversampling the positive class at the same time – SMOTE (synthetic minority oversampling technique) [5] is a popular example. We resort the latter approach to improve the robustness of both the best performing benchmark and the best performing ensemble solutions.

"Super" Ensemble/ Stacking: With complex datasets encountered in practice, there is no single approach that can be completely satisfactory in all scenarios, and hence it can be beneficial to combine them. Conventional ensemble approaches require different instantiations of the same classifier for combining, or simple bagging approaches treat all models equal by aggregating predictions through majority voting. However, we are interested in combining different models by automatically weighting them based on their usefulness for the overall performance. This is akin to building a super-model by blending different sophisticated models. Here, instead of the actual label of prediction, class-specific probabilities are obtained from each of the constituent classifiers. We concatenate those probabilities, treat them as *super-features* and build a classification model on top of them to predict the actual label. Though computationally expense, due to the need to build several models for a given dataset, such an approach is guaranteed to surpass or at least match the performance of any of the constituent models in the *Super Ensemble*.

Benchmark

Different classifier models such as decision trees, linear SVM and Naïve Bayes were fit to the data. Overall, the Decision Tree model did better than the others, especially in recall, specificity, F1 score, geometric mean and ROC. Here are the results with the actual numbers –

```
Classification Report for Linear SVM classifier
                   pre       rec       spe        f1       geo

        0.0       0.92      1.00      0.01      0.96      0.09
        1.0       0.88      0.01      1.00      0.02      0.09

avg / total       0.91      0.92      0.09      0.88      0.09

Macro AUC-ROC =  0.504045729077
Classification Report for Decision Tree classifier
                   pre       rec       spe        f1       geo

        0.0       0.93      0.98      0.23      0.96      0.47
        1.0       0.48      0.23      0.98      0.31      0.47

avg / total       0.90      0.92      0.29      0.90      0.47

Macro AUC-ROC =  0.602688401331
Classification Report for Naive Bayes classifier
                   pre       rec       spe        f1       geo

        0.0       0.93      0.91      0.24      0.92      0.47
        1.0       0.20      0.24      0.91      0.22      0.47

avg / total       0.87      0.86      0.30      0.86      0.47

Macro AUC-ROC =  0.576029951929
```

After this, ensemble methods such as Bagging, AdaBoost, RandomForest and Gradient Boosting were used on the Decision tree model to get a better benchmark score.

Bootstrap doesn't handle imbalance well, so the results are similar to the decision tree (except for precision and recall). AdaBoost has the best results in spite of the imbalanced data. It has a significantly higher AUC-ROC score than the other three ensembles and the overall higher scores as well. Random Forest and Gradient Boosting have performed better than Bagging but not as good as AdaBoost.

Here are the actual scores obtained with the ensemble methods on Decision Trees –

```
Classification Report for Bagging classifier
                   pre       rec       spe        f1       geo

        0.0       0.93      0.98      0.23      0.96      0.47
        1.0       0.53      0.23      0.98      0.32      0.47

avg / total       0.90      0.92      0.29      0.90      0.47

Macro AUC-ROC =  0.605063355109
Classification Report for Random Forests classifier
                   pre       rec       spe        f1       geo

        0.0       0.92      1.00      0.02      0.96      0.12
        1.0       1.00      0.02      1.00      0.03      0.12

avg / total       0.93      0.92      0.10      0.88      0.12

Macro AUC-ROC =  0.507611241218
Classification Report for Adaboost classifier
                   pre       rec       spe        f1       geo

        0.0       0.96      0.99      0.56      0.98      0.74
        1.0       0.84      0.56      0.99      0.67      0.74

avg / total       0.95      0.96      0.60      0.95      0.74

Macro AUC-ROC =  0.77522790583
Classification Report for GBT classifier
                   pre       rec       spe        f1       geo

        0.0       0.93      0.98      0.14      0.95      0.37
        1.0       0.44      0.14      0.98      0.21      0.37

avg / total       0.89      0.91      0.21      0.89      0.37

Macro AUC-ROC =  0.561139282633
```

III. Methodology

Data Preprocessing

The following preprocessing steps were done to the dataset –

1. All data from "Features.txt" was loaded into a NumPy array to form the input features data (X). Additionally, 7 columns from the "Info.txt" file containing coordinate information were added to the input features data. Target label from "Info.txt" was loaded into the output variable array (y).
2. Standard Scaling was done using StandardScaler from sklearn to scale the data to unit variance.
3. The data was split into training and testing sets. Splitting can be done based on ROIs or patient IDs. But since information is needed from different patients in a set, splitting was done on patients.
   Since the dataset is imbalanced, the split was not straightforward. It was done so as to ensure that there was a mix of benign and malignant patients in the data sets. This was challenging and the code shows how the split was performed.

Implementation

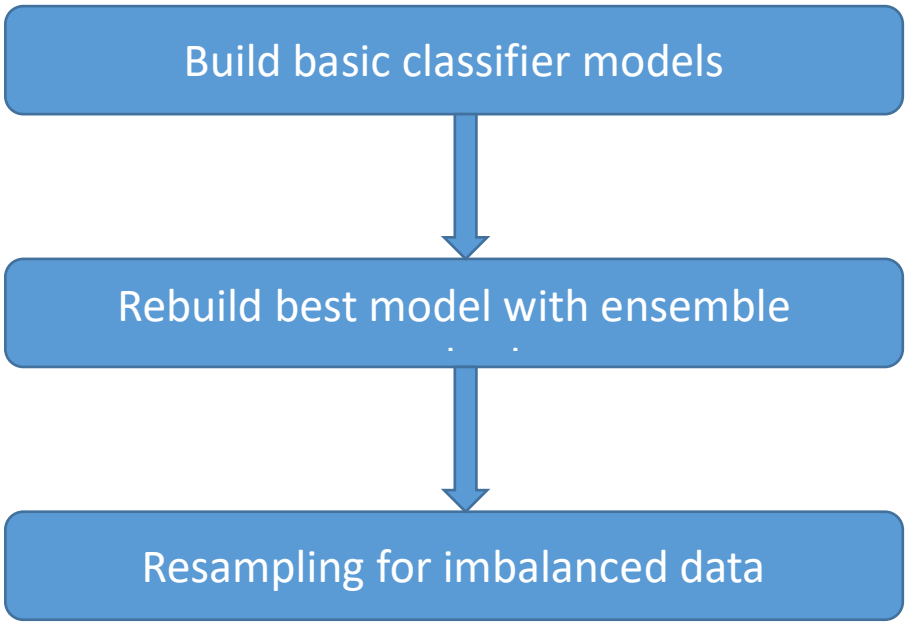

Figure 3. Implementation steps.

1. **Building classifier models**: Different classifier models such as Linear SVM, Decision Tree, Naive Bayes were fit to the data. The best performing model among these was Decision Tree and the results are described in the "Benchmark" section. Here is the code snippet –

```
classifiers = [
  LinearSVC(random_state=0),
  DecisionTreeClassifier(random_state=0, max_depth=5),
  GaussianNB()]

for name, clf in zip(names, classifiers):
  clf.fit(X_train, y_train.ravel())
  y_pred = clf.predict(X_test)
```

2. **Rebuilding models using Ensemble methods**: Ensemble methods reduce the variance of complex classifiers so that overfitting will not occur. Methods like Bagging, AdaBoost, RandomForest and Gradient Boosting were used on the best (model (Decision Tree) from Step 1 to improve classification. AdaBoost worked best and was thus used as the benchmark for the project. The results are described in the "Benchmark" section.

```
classifiers = [
  BaggingClassifier(n_estimators=10, base_estimator=DecisionTreeClassifier(max_depth=5)),
  RandomForestClassifier(n_estimators=10, max_depth=5),
  AdaBoostClassifier(n_estimators=10, base_estimator=DecisionTreeClassifier(max_depth=5) , algorithm='SAMME'),
  GradientBoostingClassifier(n_estimators=10)]

for name, clf in zip(names, classifiers):
  clf.fit(X_train, y_train.ravel())
  y_pred = clf.predict(X_test)
```

3. **Resampling for imbalanced data**: The imblearn toolset was used to implement resampling. Three models were built here and the results are shown below –

   a. SMOTE with Decision Tree

```
Classification Report for SMOTE + Decision Tree classifier
                   pre       rec       spe        f1       geo

        0.0       0.98      0.92      0.84      0.95      0.88
        1.0       0.50      0.84      0.92      0.62      0.88

avg / total       0.94      0.92      0.85      0.93      0.88

Macro AUC-ROC =  0.882381239985
```

b. SMOTE with Bagging

```
Classification Report for SMOTE + Bagging classifier
                   pre       rec       spe        f1       geo

        0.0       0.98      0.92      0.84      0.95      0.88
        1.0       0.50      0.84      0.92      0.62      0.88

avg / total       0.94      0.92      0.84      0.93      0.88

Macro AUC-ROC =  0.879822260569
```

c. Balanced Bagging (Uses balanced bootstraps during Bagging instead of random)

```
Classification Report for Balanced Bagging classifier
                   pre       rec       spe        f1       geo

        0.0       0.98      0.92      0.84      0.95      0.88
        1.0       0.50      0.84      0.92      0.62      0.88

avg / total       0.94      0.92      0.84      0.93      0.88

Macro AUC-ROC =  0.879769628991
```

Of these three models, the SMOTE with Decision Tree has slightly better metrics than the others. This makes SMOTE better performing than the benchmark models for such an imbalanced dataset.

Refinement

'Super' Ensemble/ Stacking:

A model blending technique was used to improve the results obtained using the SMOTE method. This method called stacking, trains a learner to combine the predictions of other learners. First, Decision Tree, SMOTE and AdaBoost algorithms were trained. Then two combiner algorithms – DecisionTreeClassifier and BalancedBaggingClassifier – were trained to make a final prediction using all the predictions of the other algorithms as additional inputs [6].

Both the combiner algorithms yielded results superior than the SMOTE method. The BalancedBaggingClassifier edged out DecisionTreeClassifier by a slightly higher specificity, geometric mean and ROC_AUC score.

IV. Results

Model Evaluation and Validation

The stacked model using the BalancedBaggingClassifier as the combiner algorithm was chosen to be the final model. The metrics are given below –

```
Classification Report for super ensemble with Balanced Bagging
                   pre       rec       spe        f1       geo

        0.0       0.98      0.99      0.78      0.99      0.88
        1.0       0.91      0.78      0.99      0.84      0.88

avg / total       0.97      0.98      0.80      0.97      0.88

Macro AUC-ROC =  0.885870578085
```

As seen from the metrics above, this model has an overall lead in the metric scores. This is in line with the expectation that stacked models perform better than most models.

Justification

Table 1 is a comparison of the metrics of all models. The rows in bold indicate the best performing model in that class.

Model class	Model	Metrics					
		Precision	Recall	Specificity	F1 score	Geometric Mean	ROC_AUC score
Basic Classification Models	Linear SVM	0.91	0.92	0.09	0.88	0.09	0.50
	Decision Tree	**0.90**	**0.92**	**0.29**	**0.90**	**0.47**	**0.60**
	Naïve Bayes	0.87	0.86	0.30	0.86	0.47	0.58
Ensemble models with Decision Tree	Bagging	0.90	0.92	0.29	0.90	0.47	0.61
	Random Forest	0.93	0.92	0.10	0.88	0.12	0.51
	AdaBoost	**0.95**	**0.96**	**0.60**	**0.95**	**0.74**	**0.78**
	Gradient Boosting	0.89	0.91	0.21	0.89	0.37	0.56

SMOTE	**SMOTE on Decision Tree**	**0.94**	**0.92**	**0.85**	**0.93**	**0.88**	**0.8823**
	SMOTE on Decision Tree with Bagging	0.94	0.92	0.84	0.93	0.88	0.8798
	Balanced Bagging	0.94	0.92	0.84	0.93	0.88	0.8798
"Super" Ensembles/ Stacked model	Decision Tree	0.98	0.98	0.78	0.98	0.87	0.8790
	Balanced Bagging	**0.97**	**0.97**	**0.80**	**0.97**	**0.88**	**0.8838**

Table 1. Comparison of performance metrics of all models built.

As seen from Table 1, the 'super' ensemble model with Balanced Bagging performs better than the benchmark of Decision Tree with AdaBoost. In particular, the specificity and ROC_AUC score are significantly higher in the stacked model. The performance metrics of this model are all high enough to believe that the model is a good solution for the problem discussed.

V. Conclusion

Free-Form Visualization

One characteristic in the dataset that was observed was a leakage issue. Since the data contained multiple images from each patient, if a patient had a malignant region in one of the entries, it was likely that other images of the patient might have malignant regions too. Thus there was a possibility of information leakage due to a predictive variable like Patient ID. So, a test was done to determine if leakage exists.

A dataset was created by using only Patient IDs as the input feature and the output was the target label (benign=0, malignant=1). A Decision Tree was fit to this data and the output was predicted. Even without using any of the input features (the CAD image data), the accuracy score from this experiment was a high 92.99%. The gini impurity for the Decision Tree (see Figure 4) was also very low as seen in the tree graph below.

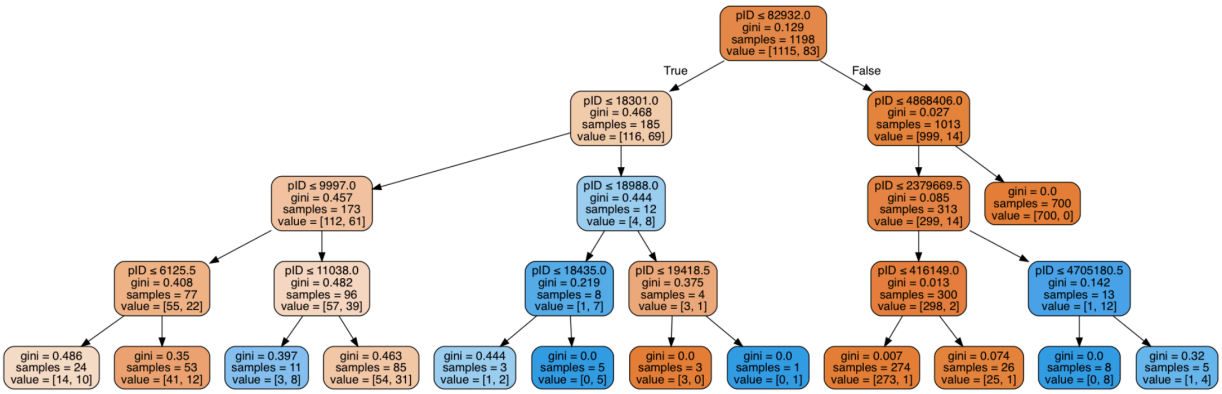

Figure 4. Decision Tree for leakage test.

It was thus discovered that the dataset has leakage due to the Patient ID variable, and was consequently not used for prediction and was only used for splitting of the dataset into train and test sets.

Reflection

The aim of the project was to construct a model that would be able to predict if a *candidate* tumor region from a breast x-ray is malignant or benign. This would help radiologists identify a breast tumor. The dataset had 124 input features corresponding to x-ray images and 1 output variable indicating if the region is malignant or benign. After preprocessing the data, several basic classifiers were fit to the training data and performance was measured using the testing data. Then, ensemble methods were run on the Decision Tree classifier to create a benchmark. In order to address the imbalance in the data, resampling methods such as SMOTE and Balanced Bagging were used. Finally, a stacked model using Balanced Bagging as the combined estimator was built and was chosen as the solution for this problem.

An interesting aspect of this project was that the dataset had a leakage issue with the “Patient ID” data. The Patient ID had predictive information about the classification of the tumor and using an experiment, it was determined that reasonably good prediction was possible using only Patient ID information and not any other relevant information about the X-ray itself. This was unfortunately a problem with the dataset, and could have been avoided by masking out or removing this field in the original dataset. The training and testing datasets used here did not use this feature.

A major difficult aspect of this project was the imbalance in the dataset. A very small percentage of the dataset was classified as ‘malignant’ tumor, and this was an issue at various stages. At the data preprocessing stage, the data needed to be split into training and testing sets such that there was a good mix of both benign and malignant tumors in both sets. At the performance measurement stage, accuracy couldn’t be used since the overall accuracy can be over 99% without detecting even a single malignant case. This imbalance was addressed using resampling methods such as SMOTE and Balanced Bagging.

The final model takes into account the challenges with the problem and the performance metrics indicate that the model is a very good solution that can be used for such a problem.

Improvement

Presently, only a transformed version of the X-ray image data was available in the dataset. One aspect that can be improved in the implementation is by having access to actual images instead of using hand-engineered features. This way, learning from the data can be accomplished using complex deep learning methods.

Another possible method to improve results can be tried by using neural networks for building the predictive model. This would be done in the future and results can be compared to the existing solution.

VI. References

[1] http://www.breastcancer.org/symptoms/understand_bc/statistics
[2] http://www.kdd.org/kdd-cup/view/kdd-cup-2008/Tasks
[3] http://www.kdd.org/kdd-cup/view/kdd-cup-2008/Data
[4] http://www.kdd.org/cupfiles/KDDCupData/2008/training_data.zip
[5] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. JAIR 16 (2002), 321–357
[6] https://en.wikipedia.org/wiki/Ensemble_learning