Kaggle Decal Final Project

Kunal Singh and Santhosh Subramanian

May 2017

1 Introduction

Breast cancer is one of the most common cancers in the United States, affecting over 250,000 people each year, and thus we wanted a data set that if worked on had the ability to affect thousands of people. This was the first driving factor to use this data set, The Breast Cancer Wisconsin (Diagnostic) Data Set, for our project. The second driving factor that made us choose this specific data set was how breast cancer personally affected me. Couple of years ago, my friends grandmother was diagnosed with stage II breast cancer, and finding whether or not the cancer was malignant or benign was not only a very expensive procedure, but also a very stressful and long one. By being able to classify whether or not the tumor itself is benign or malignant from the data set derived from the biopsy, we can detect cancer earlier thus saving patients both time and money.

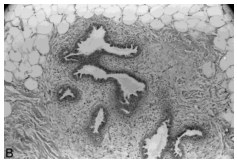

Figure 1: Digitized Image

The data set that we worked with was the The Breast Cancer Wisconsin (Diagnostic) Data Set. "The features are derived from a digitized image of a fine needle biopsy of a breast mass (Figure 1[1]). The features describe characteristics of the cell nuclei present in the image."[2]. There were 569 instances which features were derived from, and out of these instances 357 of them were benign and 212 of them were malignant. For each instance, 14 features were derived. Furthermore, for each feature, the mean, standard error, and "worst" or largest (mean of the three largest values) were computed. Using this data, our main goal was to build machine learning models to classify these breast mass instances as benign or malignant.

[1] https://dollar.biz.uiowa.edu/~nstreet/research/cc97_02.pdf

[2] http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.names

2 Data Processing

The Breast Cancer Wisconsin (Diagnostic) Data Set repository included three data sets; however, we used the second data set, wdbc.data, because it contained the most features and had the most recent data collected.

These are the some of the included features:

- Radius (mean of distances from center to points on the perimeter)
- Texture (standard deviation of gray-scale values)
- Perimeter
- Area
- Smoothness (local variation in radius lengths)
- Compactness (perimeter^2 / area - 1.0)
- Concavity (severity of concave portions of the contour)
- Concave Points (number of concave portions of the contour)
- Symmetry
- Fractal Dimension ("coastline approximation" - 1)

Featurization:

The first step in the data processing was that we separated the whole data set into three smaller sets, one with the mean of the features, one with the standard error of the features, and one with mean of the three largest values (worst). This was done so that we could find the correlations between features in order to remove unnecessary features in the data set in order to improve accuracy and reduce training time. We found that the *radius*, *perimeter*, and *area* are highly correlated, and also that *compactness*, *concavity* and *concave points* are also highly correlated.

The second step we did was create a training dictionary based on the correlations found above. The dictionary includes *columns_total*, *columns_mean*, *columns_se*, and *columns_worst*.

The third step we did was testing for feature importance (bulk of our featurization). We used a random forest classifier on the whole data set to predict which features were most important. In general, standard error features are of least important. Thus, *perimeter_se*, *radius_se*, *area_se* are taken out. *compactness_worst* taken out because it is highly correlated with *concave_points* and *concavity*. However, *concavity_mean* and *concavity_worst* are included because they still have high importance. Furthermore, *area* and *radius* both included because both of high importance.

3 Data Modeling

For each machine learning model that we developed, hyper parameter tuning was done. To do this, we first created a parameter grid and then applied the grid search function on the dictionary. We also preformed cross-validation for each machine learning model. We did this by first creating a cross-validation function and then applying it for each model. We decided to use four machine learning models:

SVM:

A support vector machine is a supervised learning algorithm that "outputs an optimal hyperplane which categorizes new examples."[3] The advantages of using a SVM classifier is that:

- Memory Efficient as it can handle thousand plus support vectors.
- SVM's are versatile as we can specify our own custom kernels when building the classifier. For example, we did a grid - search using RBF and linear kernels.
- Resistant to over-fitting as once a hyperplane is found, changes in the data do not greatly affect the SVM as the hyperplane identifies boundaries efficiently.

Random Forest:

"A Random Forest consists of a collection or ensemble of simple tree predictors, each capable of producing a response when presented with a set of predictor values. This response takes the form of a class membership."[4]In layman's terms, a random forest is an ensemble of decision trees The advantages of using a random forest is that:

- Runs efficiently on Large Data Sets
- Gives estimates of what features are most important. We used random forests in our project to find which features were most important.
- Does not over fit
- Helps balance error in data sets

Logistic Regression:

Logistic Regression is basically regression that is not linear. It "involves a

[3] http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

[4] http://www.statsoft.com/Textbook/Random-Forest

more probabilistic view of classification."[5] The advantages of using this model is that:

- The model does not assume the relationship between variables is linear.
- Low Variance
- The model is robust where variables do not have to normally distributed.

Decision Treees:

"Decision Trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features."[6]

- Very easy to read and interpret
- Doesn't require much data
- Able to handle both numerical and categorical data
- "Performs well even if its assumptions are somewhat violated by the true model from which the data were generated."[7]

Neural Network:

"A neural network is a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs."[8] This system is comprised of organized layers which contain nodes (neurons) which contain activation functions. We built our neural network (sequential model) with five layers: a dense layer (30 neurons), a relu activation layer (30 neurons), another dense layer (30 neurons), even another dense layer (128 neurons), a softmax activation layer (10 neurons), and finally a another dense layer (10 neurons). Finally, we implemented a sparse categorical cross entropy loss function with a sgd optimizer. The advantages of neural networks are: [9]

- Neural networks have the ability to detect all possible interactions between predictor variables
- Neural networks can be developed using multiple different training algorithms
- Neural networks can implicitly detect complex nonlinear relationships between independent and dependent variables

[5] http://courses.washington.edu/css490/2012.Winter/lecture_slides/05b_logistic_regression.pdf

[6] http://scikit-learn.org/stable/modules/tree.html

[7] http://scikit-learn.org/stable/modules/tree.html

[8] http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html

[9] http://www.sciencedirect.com/science/article/pii/S0895435696000029

4 Comparison of Models

Table 1: Machine Learning Models

	SVM	**NeuralNet**	**DecisionTree**	**RandomForest**	**LogisticRegression**
Accuracy	97.400%	98.246%	95.600%	98.200%	L1 - 98.200% L2 - 99.100%
CVS1	91.228%	N/A	91.228%	92.105%	L1 - 95.614% L2 - 93.860%
CVS2	94.298%	N/A	92.544%	93.421%	L1 - 96.491% L2 - 96.053%
CVS3	95.029%	N/A	93.567%	95.029%	L1 - 97.076% L2 - 96.491%
CVS4	95.175%	N/A	94.518%	95.614%	L1 - 97.149% L2 - 96.272%
CVS5	95.255%	N/A	94.729%	95.960%	L1 - 97.011% L2 - 95.956%

—— CVS: Cross Validation Score

The model that preformed the best was the Logistic Regression model, then the Neural Network, then the Random Forest model, then the SVM, then finally the Decision Tree. The Logistic Regression model was the best because it had the highest accuracy while having a really good cross validation score.

5 Discussion

The model that had the highest accuracy rate was the Logistic Regression; however, it did not have the highest cross validation scores. All models did have an extremely high accuracy, all above 95%. The models (other than the neural network) also did have high cross validation scores - none below 90%. Since the models we built have high validation scores, there is minimal need to worry about over-fitting. Thus, we could see these models working effectively on data sets with the same features. However, if our cross-validation scores were low, then there would be a need to worry about over-fitting.

There are a lot of implications with the project. Like stated before, since our models had a high accuracy and high cross validation scores, we could use these models on different breast mass data sets with the same features.

For our project, nothing really went wrong. However (though still high accuracy), our SVM model could have overfit as it had relatively lower cross-validation of all the models. Another thing that caused us trouble was getting

the neural network to a high accuracy. Our training data was initially formatted incorrectly, and it took a while to add the right layers in the right positions. For future work, we can train and test the neural networks using different training/test splits.

6 Conclusion

Since we were given a lot of freedom with the project including being able to choose our own data set and models, it was very interesting to see what we could do with the data given. The first parts of the project were tedious and dull. These steps include prepossessing and processing the data. The only interesting part at this time was when we were processing the data for important features. Building the machine learning models was not too difficult; however, hypertuning the parameters and performing cross - validation was very tedious. The results we achieved from the models was mind-blowing as we achieved such high accuracies with respective high cross validation scores. If we had more time, one thing that we would have implemented is preforming k-fold cross validation on the neural network. Another thing that we would have done if we had more time would be changing the size of the training/testing sets for the neural network. We believe that this project was well done as it tested on most of material we learned in the class, and we had to implement much of the material from the lectures into the project. The biggest challenge of the project was building the neural network as it is the most complex model that we chose. Similarly, processing the data was also a big challenge as the data set was very large. Though it took some time, we overcame both problems pretty easily. We kept guessing and checking and learnt from what was wrong from the check before. Using this method, we were able to pinpoint what we did wrong and were able to fix the problem quickly.