Assessment of Clinical Predictors for Breast Cancer Diagnosis Using Statistical Learning Methods

Fandi Chang, Jasmine Wang, Lin Zou, Yezhi Pan

December 3, 2020

1 Introduction

Breast cancer is the most common cancer in American women. The average risk of a woman in the United States developing breast cancer is about 13% (About Breast Cancer, n.d.). This means that there is a 1 in 8 chance of developing breast cancer. Breast cancer is also the second leading cause of deaths due to cancer in women. Since 2007, death rates have been steady in women younger than 50, but have continued to decrease in older women. From 2013 to 2017, the death rate decreased by 1.3% per year. These decreases are believed to be the result of identifying breast cancer earlier through screening and increased awareness, as well as the recent advancements in treatment. It is important to understand that most breast lumps are benign. Non-cancerous breast tumors are abnormal growths, but they do not spread to other parts of the body. Hence, they are not life threatening, but some types of benign breast lumps can increase the risk of developing breast cancer. Regardless, an early detection of lumps is beneficial to the treatment process and the overall well-being of the patient.

When a lump is detected early, the next question is whether it is benign or malignant. One procedure to answer this question is to perform a fine needle aspiration (FNA) biopsy of the breast. During this quick and non-invasive procedure, a very thin, hollow needle attached to a syringe is used to withdraw a small amount of tissue or fluid from a suspicious area and is checked for cancer cells. In some cases, it is possible to diagnose the patient on the same day. It is evident that FNA is a useful procedure for the early detection of breast cancer.

Taking this into consideration, we want to know what specific aspects of the FNA results are associated with cell diagnosis. To answer this question, we used the Breast Cancer Wisconsin dataset ($N = 569$), which contains features computed from a digitized image of a FNA of a breast mass (Breast Cancer, 1995). The features describe characteristics of the cell nuclei present in the image, which includes the 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 ($\frac{\text{perimeter}^2}{\text{area}-1}$),area concavity (severity of concave portions of the contour), concave points (number of concave portions of the contour), symmetry, fractal dimension ("coastline approximation"-1). The mean, standard error, and "worst" or largest (mean of the three largest values) of these dimensional features were computed for each image, resulting in 30 features of interest. We implemented a variety of machine learning methods, such as principal component analysis and regression, logistic regression, K Nearest-Neighbors, and decision trees, to describe the association between the features of interest and cell diagnosis, and to predict cell diagnosis using the most important features.

2 Methods

2.1 Logistic Regression

2.1.1 Variable Selection

Before we build our model, we would like to check for correlations. Since we have 32 prediction variables in our original data set, and for logistic regression, multicollinearity can cause issues such as yielding solutions that are wildly varying and possibly numerical unstable. It is also not ideal to include all 32 variables into our model, and so it is very important for us to remove highly correlated predictors to make our model and analysis more robust. In order to have a general overview of the correlational relationship between variables, we create a correlation plot (Figure 1) indicating correlation values by different color from light to deep. According to Figure 1, we remove the highly correlated variables with values bigger than 0.9, using the *caret* package and *findCorrelation* function in R. After solving multicolinearity issue, we now have 10 variables shorter and only have 22 variables preparing for building our logistic regression model.

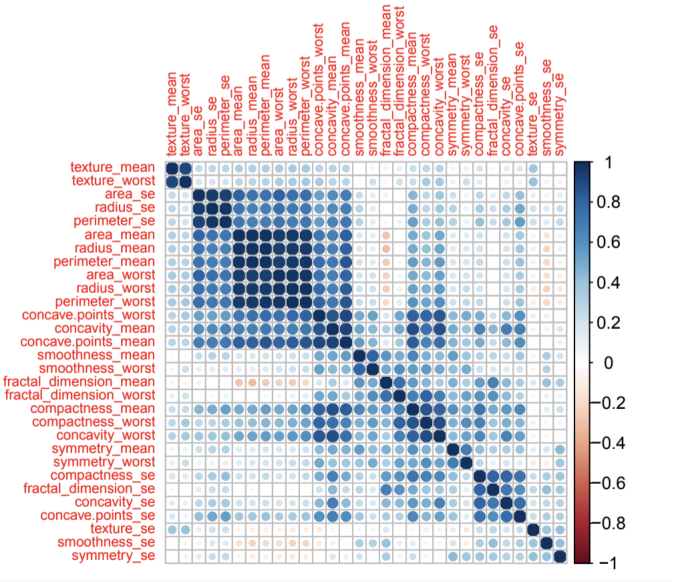

Figure 1: Correlation Plot for Predictors

2.1.2 Statistical Background

Logistic Regression is a generalized form of linear regression that is used to examine the association of independent variables (categorical or continuous) with dichotomous dependent variable (Nourelahi et al., 2019). For multiple linear regression where having vector $X = (x_1, x_2, ..., x_m)$ and the output Y with equation $Y = \beta_0 + \sum_1^m \beta_i x_i$, where β_i indicates the model coefficients and are estimated using least squares method. For our binomial logistic regression, instead of having dependent variable being continuous, our output has binary result. Hence, in building a model, we first calculate the odds of the event, and define it as $\frac{\pi(x)}{1-\pi(x)}$, where $\pi(x)$ represents the probability of success. The link function $g(\mu_i) = \sum_{j=1}^p \beta_j y_{ij}$ is used to connect the systematic and random components. In other words, we have

$$g(E(R_i)) = \sum_{j=1}^{p} \beta_j y_{ij}$$

$$g(\pi_i) = \sum_{j=1}^{p} \beta_j y_{ij}$$

$$\log(\frac{\pi_i}{1-\pi_i}) = \sum_{j=1}^{p} \beta_j y_{ij}$$

where the probability can be derived as $\pi = \text{logit}^{-1}(Y\beta) = \frac{e^{Y\beta}}{1+e^{Y\beta}}$ Here, $0 \leq \pi \leq 1$, where the equality to boundaries happens when $Y\beta \to \pm\infty$, and $\Omega = \frac{P(R=1|Y)}{P(R=0|Y)} = e^{Y\beta}$. The estimated intercept coefficient, calculated by maximum likelihood method (Nourelahi et al., 2019), is the log odds when all the predictors are equal to zero. The estimated β coefficients are the log odds ratios and can be interpreted as odds ratios after exponentiation. An estimated 95% confidence interval for β can be computed as $\hat{\beta} \pm 1.96SE(\hat{\beta})$, and an estimated 95% confidence interval for e^{β} can be computed as $e^{\hat{\beta} \pm 1.96SE(\hat{\beta})}$. Since the effect is an odds ratio, logistic regression is a multiplicative risk model.

2.1.3 Model Evaluation

Our data set is divided into training and testing sets, where training set contains 80% of our data and testing set has the rest 20%. We use 10-fold cross validation to reduce the error caused by bias and variance (Nourelahi et al., 2019). Applying 10-fold cross validation as the resampling method using *trainControl* function, we build a logistic regression model based on the training set. Then we obtain our predicted results by using *predict* function applying our model on testing set. A confusion matrix is created to generate a overview for our results.

2.2 Principal Component Analysis with Logistic Regression

Principal component analysis (PCA) finds linear combinations of variables that best explain their covariation structure. In particular, the method is useful for emphasizing variation and highlighting strong patterns in a dataset. The main idea is that each of the n observations lives in a p-dimensional space, but not all of these dimensions are equally interesting. PCA finds a small number of dimensions that are as interesting as possible, where the concept of interesting is measured by the amount that the observations vary along each dimension.

Let $\boldsymbol{X} = (X_1, X_2, ..., X_p)$ have the covariance matrix $\boldsymbol{\Sigma}$ with eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p \geq 0$. Consider forming new variables $Y_1, ..., Y_p$ by taking p different linear combinations of the X_j variables:

$$
\begin{aligned}
Y_1 &= \boldsymbol{a}_1^T \boldsymbol{X} = a_{11}X_1 + a_{12}X_2 + ... + a_{1p}X_p \\
Y_2 &= \boldsymbol{a}_2^T \boldsymbol{X} = a_{21}X_1 + a_{22}X_2 + ... + a_{2p}X_p \\
&... \\
Y_p &= \boldsymbol{a}_p^T \boldsymbol{X} = a_{p1}X_1 + a_{p2}X_2 + ... + a_{pp}X_p
\end{aligned}
$$

where $\boldsymbol{a}_k = (a_{k1}, a_{k2}, ..., a_{kp})$ are the kth loadings for the kth principal component (Friedman, Hastie, & Tibshirani, 2001). Since $Y_k = \boldsymbol{a}_k^T \boldsymbol{X}$, it has the following properties: $Var(Y_k) = \boldsymbol{a}_k^T \boldsymbol{\Sigma} \boldsymbol{a}_k$ and $Cov(Y_k, Y_l) = \boldsymbol{a}_k^T \boldsymbol{\Sigma} \boldsymbol{a}_l$. Thus, the principal component loadings are the uncorrelated linear combinations $Y_1, ..., Y_p$, whose variances are maximized. A PCA solution can be achieved through eigenvalue decomposition. We can express the population covariance matrix $\boldsymbol{\Sigma}$ as

$$
\boldsymbol{\Sigma} = \mathrm{UDU}^{\boldsymbol{T}} = \sum_{k=1}^{p} d_k \boldsymbol{u}_k \boldsymbol{u}_k^T
$$

where $\boldsymbol{U} = (\boldsymbol{u}_1, ..., \boldsymbol{u}_p)$ contains the eigenvectors of $\boldsymbol{\Sigma}$ and $\boldsymbol{D} = \boldsymbol{diag}(d_1, ..., d_p)$ contains the eigenvalues of $\boldsymbol{\Sigma}$. The PCA solution is obtained by setting $\boldsymbol{a}_k = \boldsymbol{u}_k$, for $k = 1, ..., p$. Since the new variables $Y_1, ..., Y_p$ are obtained by taking the linear combinations of the original variables $X_1, ..., X_p$, it follows that $Y_1, ..., Y_p$ has the same total variances as $X_1, ..., X_p$. Moreover, the proportion of the total variance accounted for by the kth principal component is $R_k^2 = \frac{d_k}{\sum_{j=1}^{p} d_j}$. It follows that the proportion of the total variance accounted by the first r principal component is $\sum_{k=1}^{r} R_k^2 = \sum_{k=1}^{r} \frac{d_k}{\sum_{j=1}^{p} d_j}$. Therefore, if $\sum_{k=1}^{r} R_k^2 \approx 1$ for some $r < p$, we do not lose much information by transforming the original p variables into r new principal component variables.

To perform principal component logistic regression, we fit a logistic regression model using the uncorrelated linear combinations $Y_1, ..., Y_p$ obtained from PCA as our new predictors. The mean and variance of the response can be computed as $E(R_i) = \pi_i$ and $Var(R_i) = \frac{\pi_i(1-\pi_i)}{n}$.

2.3 Decision Trees

2.3.1 CART VS. C5.0

The next model we implemented in this paper is decision tree. Decision trees are algorithm-driven, recursive partitioning supervised machine learning models that split the predictor space, X_1, X_2, $X_3 ... X_p$, into J simple, non-overlapping regions, $R_1, R_2 ... R_J$, and make the same prediction for each observation that falls into region R_J (James et al., 2017). Decision tree models have several advantages as compared with other statistical learning methods. First, recursive partitioning models tend to have high interpretability and are more intuitive to understand. Second, these models are not parametric, and do not assume certain functional relationship between response variable and predictor variables. Third, decision trees are good at handling missing values and large feature numbers, but this is not related to our paper because our dataset do not have missing values. Fourth, decision trees can prioritize a metric by putting more weights on misclassification costs, and in our case this is very useful because diagnosis models prefer high sensitivity over specificity or overall accuracy. Despite all the advantages, simple decision tree models are only implemented here as a reference model because they are non-robust, as a small change in the input can result in an entirely different estimated model, thus these kind of models should not be referenced to alone in medical application such as cancer diagnosis. We should consider using tree-based methods like random forest if higher model stability is required but this will result in the loss of interpretability.

In our paper, we are interested in predicting tumor types so our response variable is categorical and we will focus on classification trees. Classification tree algorithms choose the splitting variable at each node, the criterion for splitting at each node, and the criterion for terminating a node (Agresti, 2012). Each algorithm has its own criterion for building a tree, but one thing in common is that the impurity is measured at each node and the algorithms optimize purity to make splits. The most intuitive choice of measurement of impurity in classification trees is the proportion of the training observations that do not belong to the most common class, which is also known as the misclassification error rate:

$$E = 1 - max_k(\hat{p}_{mk})$$

where $\hat{p}_{mk}$ stands for the proportion of class kth observations in node m and has the expression:

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(Y_i = k)$$

where N_m stands for the number of observations in region R_m. In practice, there are two other measures of impurity that are popular because misclassification error rate is not sufficiently sensitive (James et al., 2017). The first is the Gini index:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

that measures total variance across the K classes, and a small Gini index indicate that the majority or all observations in node m are from a single class. The second measure is entropy:

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

and similarly, a small entropy value indicates purity at node m.

Specifically, we will use and compare two representatively distinct types of tree-based classification algorithms: the classification and regression tree model (CART) developed by Breiman and the C5.0 algorithm developed by Quinlan. CART uses Gini index as the splitting criterion whereas C5.0 uses entropy. Since the two measures are quite similar numerically, what makes the two algorithms really different is the fact that CART is a pre-pruning model and C5.0 is post-pruning. C5.0 algorithm first grows an over-fitting tree until the nodes cannot be split any further, and re-evaluate the value of leaves to decide whether they significantly contribute to the prediction model. Then the algorithm prunes the splits based on binomial confidence limit. On the other hand, CART makes binary splits by minimizing the Gini Index until stopping rules control the trees to stop growing. The CART algorithm prunes trees using the cost-complexity algorithm to tune the model.

2.3.2 Parameter Tuning

We use three methods to tune the decision tree models: tree pruning in CART, weighting misclassification error cost in C5.0, and boosting in C5.0. First of all, tree pruning is the process of removing excessive splits in the tree-based models and reducing the risk of over-fitting the data. The CART algorithm prunes trees by optimizing the cost-complexity criteria

$$R_a(T) = R(T) + \alpha \cdot |f(T)|$$

where $R(T)$ is the training error, α is the complexity parameter, and $f(T)$ is a function that returns the number of terminal nodes in tree T. For classification trees, $R(T)$ is the misclassification error rate which has the expression:

$$R(T) = \sum r(t) \cdot p(t) = \sum R(t)$$

in which $r(t)$is the misclassification error rate illustrated in the previous section, $p(t)$ is the proportion of observations in a node takes in the dataset, and $\sum R(t)$ represents the sum of misclassification errors at all terminal nodes. The cost-complexity pruning obtains a function of α which represents a sequence of best subtrees (James et al., 2017). Then the algorithm uses 10-fold CV to choose a α that minimizes the average error, and returns to the sub-tree that corresponds to the chosen value of

α. The second method is to inflate the cost for a specific error to lower the corresponding error rate. In our study, our goal is diagnosis prediction with the application in breast cancer, therefore, failing to diagnose a malignant tumor is much more costly than identifying falsely a benign tumor as malignant. As a result, high sensitivity is preferred over specificity or overall accuracy, and this method would help us to improve our model performance in high sensitivity. The third tuning method is to use boosting in the C5.0 algorithm. Boosting is a slow but powerful learning process that grows trees sequentially and learns from previous trees. Boosting assigns a weight to previously mis-classified samples and applies the weight to the calculation of information gain (entropy). In the C5.0 algorithm, boosting iterations can be set via the *trials* argument.

2.4 K Nearest-Neighbor

K Nearest-Neighbor method (KNN) is one of the supervised machine learning algorithm (Friedman, Hastie, & Tibshirani, 2001). In other words, the predicted outcome depends on both predictors and outcomes in training data set. The basic idea of KNN is to assume similar inputs have similar outputs. That is to say, it uses the observations that are most similar to a given set of predictors to form predictions on the response variable. There are multiple metrics for similarity such as Euclidean distance, Hamming distance, and Manhattan distance. We used Euclidean distance in this study because all predictors are continuous and it is easy to compute and interpret. If the response variable is continuous, the model can be written as $\hat{Y}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i$, where $N_k(x)$ is the neighborhood of x defined by the k points x_i that have the smallest Euclidean distance. In other words, after we find the k observations with x_i closest to x, the prediction on this new data point will be the average of their responses. Similarly, if the response variable is categorical, the majority vote rule is applied here. Therefore, the class of this new data point will be assigned to the class that has the largest proportion in k responses.

KNN only has one single parameter which is the number of neighbors k. The choice of k has a direct impact on the model complexity. When k is small, the decision boundary is unstable which implies high variance. When k is large, the variance will decrease and the bias will increase. So, it is critical to find a suitable k to balance the variance and bias. There is no pre-defined method to choose k. In this study, we use 10-fold cross validation and specified k as the tuning parameter. Then, we choose the k with the smallest cross validation error.

KNN has several advantages. First, it is non-parametric which means it does not rely on any assumptions about the underlying data. Hence, KNN is more flexible than some parametric methods like linear regression. Additionally, KNN is easy to implement and interpret compared to other machine learning algorithms. On the other hand, KNN also have some disadvantages. First, since KNN is non-parametric, the model predictions will depend heavily on the data set which will result in higher variance than other models like linear regression. Moreover, KNN might not work well on high-dimensional data. The size of data space will increase exponentially as the number of predictors increases. As a result, the distance between points will be farther and farther apart.

3 Results

3.1 Logistic Regression

When fitting the logistic regression model, though some variables do not show very small p-value (smaller than 0.05), we do keep them into our model to have a better performance (See Figure 5 in Appendix). The odds ratio of each feature indicates how each change in the features could alter the chances of survivability. For instance, we say that for variable *symmetry_worst*, for a one-unit increase in this variable, we expect an increase in the log odds of chance of survival by 2.9932 when keeping other predictors the same.

3.2 Principal Component Analysis with Logistic Regression

As previously mentioned, many of the features are highly correlated with each other. Usually, highly correlated variables would cause the problem of multicollinearity when fitting a logistic regression model,

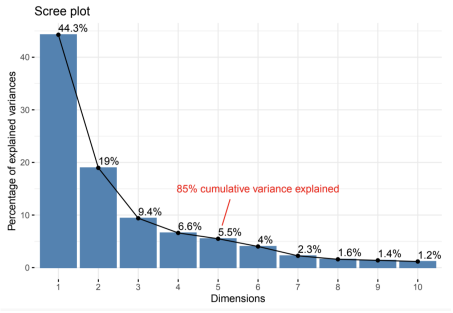

Figure 2: The Scree Plot of PCs

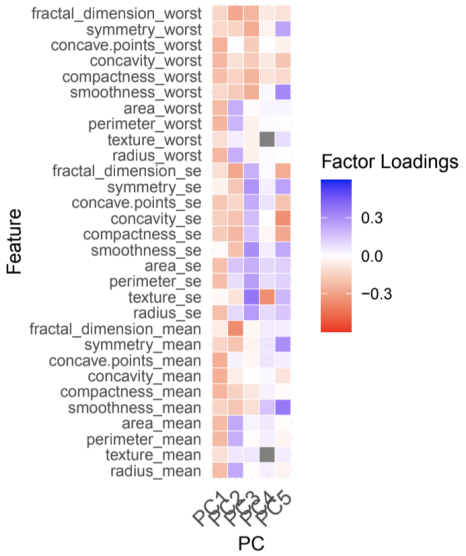

Figure 3: Heatmap of PC Loadings

but with a dimension reducing method such as PCA, we are able to obtain uncorrelated linear combinations $Y_1, ..., Y_p$ as our new predictors. We can observe from our scree plot (Figure 2) that the first five principal components, PC1 (44.3% variance explained), PC2 (19% variance explained), PC3 (9.4% variance explained), PC4 (6.6% variance explained), and PC5 (5.5% variance explained), account for approximately 85% of the total variance explained. The percentage of variance explained does not change much after the fifth principal component, so we decided to retain only the first five principal components.

Figure 3 is a heatmap of the PCA loadings, and we can observe that none of the loadings are large ($>$ 0.80). The cells with high PC1 scores have low values for all the features, which means that these cells are most likely benign. The cells with high PC2 scores have high area, perimeter, texture, and radius features. The cells with high PC3 scores have high standard error values for all the features and low "worst" values for fractal dimension, symmetry, concave points, concavity, compactness, and smoothness. The cells with high PC4 scores have low values for the texture feature. Lastly, the cells with high PC5 scores have high symmetry and smoothness features, and the standard errors for area, perimeter, texture, and radius are high as well. More research must be conducted on the biology of breast cancer and the FNA biopsy procedure to summarize or rename these five principal components into more interpretable results.

Let us consider fitting the logistic regression model with diagnosis as the response and the five principal components as the predictors. That is,

$$log(\frac{\pi}{1-\pi}) = \alpha + \beta_1 \text{PC1} + \beta_2 \text{PC2} + \beta_3 \text{PC3} + \beta_4 \text{PC4} + \beta_5 \text{PC5}$$

We obtained the results by using the validation set approach to train the logistic regression model on 80% of the data and testing the model on the remaining 20% of the data (See Figure 6 in Appendix). Although the model provides good predictions, the estimated coefficients are very difficult to interpret due to the additional layer in interpretation from using the PC scores are predictors. The intercept can be interpreted as the log odds, $log(0.64) = -0.4462871$, of being malignant when all the other predictors are zero. The estimated coefficient for the intercept is not statistically significant at $\alpha = 0.05$. All of the estimated coefficients for the principal component terms are statistically significant at $\alpha = 0.05$. A one unit increase in PC1 will increase the odds of being malignant by 0.05 times, while holding all other PC terms constant. A one unit increase in PC2 will increase the odds of being malignant by 7.27 times, while holding all other PC terms constant. A one unit increase in PC3 will increase the odds of being malignant by 0.44 times, while holding all other PC terms constant. A one unit increase in PC4 will increase the odds of being malignant by 0.39 times, while holding all other PC terms constant. A one unit increase in PC5 will increase the odds of being malignant by 6.48 times, while holding all other PC terms constant.

Our interpretation of the PCA loadings and the estimated coefficients from the logistic regression model are consistent. From PC2 and PC5, it appears that cells with high values of symmetry, smoothness, area, perimeter, texture, and radius are more likely to be malignant. Whereas from PC1, PC3, and PC4, it appears that cells with low values for all the features or high standard error values for all the features are less likely to be malignant. In particular, PC4 suggests that cells with low values for the texture feature are less likely to be malignant.

3.3 Decision Trees

Based on the decision tree methodology demonstrated above, we run the models in R. We use *rpart* package for CART models and *C50* package for C5.0 models. When comparing models, we use 10-fold cross validation on the entire dataset to compare the classification error rate and sensitivity. After comparing the tree model performance, we split the dataset and use 80% as training set and 20% as test set to interpret the model. We first fit a CART model to predict the diagnosis type with all 30 features, and prune the model in accordance with the optimal complexity parameter: the cross validated discrepancy between observed classification and predicted classification.

The result shows that post-pruned CART is the same as the pre-pruned CART, with a CV classification error rate of 0.11 and sensitivity of 0.82. The CART model chooses *concave.points_worst* as the root node and has 5 nodes, 4 predictor variable in total. We next fit a C5.0 model without any specification and this model out-performed the CART model with a lower CV classification error rate of 0. 09 and a higher sensitivity of 0.88.

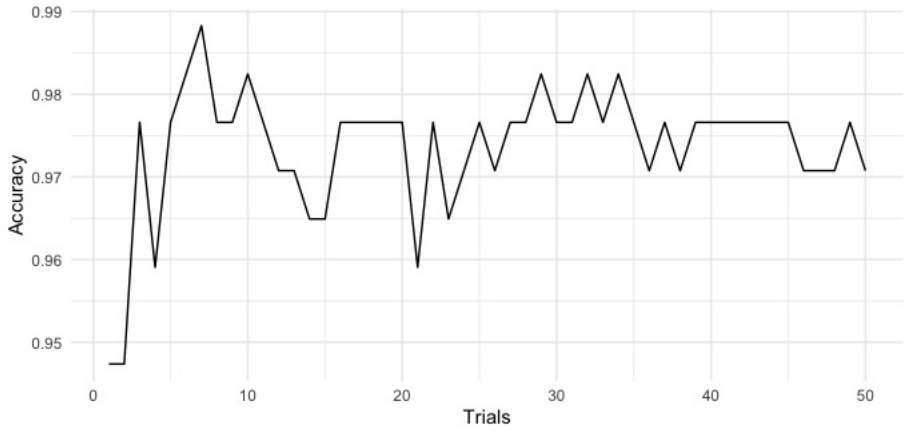

Figure 4: Accuracy of Different Trials of the Cost-Sensitive C5.0 Model

Then we tune the C5.0 model by assigning the classification error matrix to the cost argument, in order to reduce the false negative rate due to our goal of early prediction of cancer. After adjusting the C5.0 model to be cost-sensitive, the overall prediction error rate increases from 0.09 to 0.12 but the sensitivity also increases from 0.88 to 0.91 as desired. Our next step is to find the best boosting iterations of the adjusted C5.0 model. Figure 4 was generated by running 50 trials of boosting iterations in the C5.0 algorithm and recording corresponding test set classification accuracy of each number of trials. The number with highest test set prediction accuracy is 7, hence we set the *trials* argument in the C5.0 equal to 7. The CV result indicates that the classification error rate, 0.08, is the lowest among all four models, with a small trade-off in sensitivity, which drops from 0.91 to 0.90.

From our results, we conclude that the C5.0 cost-sensitive model with 7 boosting iteration is the best decision tree model among the four. In the model, there are 19 nodes with 9 predictor variables, and the *perimeter_worst* is the root node. This model contains more nodes and predictors than the CART model and does not contain variable *concave.points_worst* which is of the most importance in the CART model.

3.4 KNN and Model Comparison

For KNN, by applying 10-fold cross validation, k equals 13 has the smallest cross validation error rate. For model comparison purpose, we compute sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), and prevalence of all models via 10-fold cross validation. The results are shown in Table 1. The numbers are presented in percentage.

Table 1: Comparison of Model Performance

Model	Error Rate	Sensitivity	Specificity	PPV	NPV	Prevalence
Logistic Regression	4.42	97.62	94.37	91.11	98.53	37.17
PCA with Logistic Regression	2.64	98.10	96.33	97.75	96.54	37.23
KNN	6.68	87.58	96.79	93.72	93.04	37.25
Decision Tree-CART	10.62	82.22	93.59	89.95	90.09	37.26
Decision Tree-C5.0	9.08	87.82	92.83	88.21	92.86	37.26
Decision Tree-C5.0 Adjusted	11.91	90.61	86.64	80.89	94.01	37.26
Decision Tree-C5.0 Tuned	7.68	90.18	93.65	89.71	94.17	37.26

According to Table 1, the prevalence of breast cancer is around 37% regardless of models. Principal component analysis with logistic regression model has the lowest error rate of 2.64%, highest sensitivity of 98.10%, and a high specificity of 96.33%. Logistic regression also has a relatively low error rate of 4.42%, a high sensitivity of 97.62%, and specificity of 94.37%. On the other hand, KNN has the highest specificity (96.79%) while the accuracy (6.68%) and sensitivity (87.58%) are both lower than logistic regression and principal component analysis with logistic regression model. Additionally, C5.0 cost-sensitive model with 7 boosting iteration has higher sensitivity (90.18%) than KNN while the accuracy and specificity score lower than other models.

In general, logistic regression and principal component analysis perform better than tree-based models and KNN in terms of the accuracy of predicting the breast cancer diagnosis. It is hard to find a model that scores the highest on all possible metrics. Here, we choose the model based on error rate, sensitivity, and specificity. As a result, principal component analysis with logistic regression is the best model in predicting breast cancer diagnosis.

4 Conclusions

Our study examines four prediction models' performance in early breast cancer diagnosis and determines which feature sets provide most predictive values in our models. When comparing model performance, it is important to use the criteria that applies to the study setting. Within the scope of our study, our main criteria is low false negative rate but there could be scenarios where other metrics should be put more emphasis on. Furthermore, we observe the interpretability-prediction trade-off throughout our study: PCA boosts the model prediction performance but hinders interpretation of feature behavior in the logistic regression model; C5.0 models have more complex rulesets than CART models but have higher classification accuracy in the meantime. The interpretability-prediction trade-off plays a crucial part in our study because not only do we want to build a classification model with high sensitivity but we also want to assess feature significance in the models, so that we can have a more intuitive understanding on how features are associated with the outcome, and assist doctors in clinical decision-making and improve the early diagnosis rate of breast cancer.

References

About Breast Cancer: Breast Cancer Overview and Basics. (n.d.). Retrieved December 02, 2020, from https://www.cancer.org/cancer/breast-cancer/about.html

Agresti, A. (2012). Categorical Data Analysis, 3rd edition.

Breast Cancer Wisconsin (Diagnostic) Data Set.(1995). Retrieved from https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The Elements of Statistical Learning. New York: Springer Series in Statistics.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning with Applications in R, 7th printing.

Nourelahi, M., Zamani, A., Talei, A., Tahmasebi, S. (2019). A Model to Predict Breast Cancer Survivability Using Logistic Regression. Middle East Journal of Cancer, 10(2), 132-138. doi: 10.30476/mejc.2019.78569.

Appendix: Essential Visualizations

```
Call:
NULL

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.07673  -0.04544  -0.00707   0.00001   2.78859

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                   2.9615     1.4170   2.090   0.0366 *
perimeter_mean               -4.0185    11.2767  -0.356   0.7216
area_mean                    -3.1805    11.6981  -0.272   0.7857
smoothness_mean              -0.4534     1.2014  -0.377   0.7059
`\\`concave points_mean\\``   5.0471     2.7214   1.855   0.0637 .
symmetry_mean                -1.2655     0.9205  -1.375   0.1692
fractal_dimension_mean       -1.0406     1.2247  -0.850   0.3955
radius_se                    -2.7128     5.7552  -0.471   0.6374
area_se                      11.7966    10.5621   1.117   0.2640
smoothness_se                 0.6908     0.8771   0.788   0.4310
compactness_se               -0.6558     1.6669  -0.393   0.6940
concavity_se                 -1.3142     1.7186  -0.765   0.4445
`\\`concave points_se\\``     0.5887     2.0441   0.288   0.7733
symmetry_se                  -1.3766     1.1472  -1.200   0.2302
radius_worst                 -4.4886    12.6320  -0.355   0.7223
area_worst                   15.1260    16.3726   0.924   0.3556
smoothness_worst             -0.1627     1.2509  -0.130   0.8965
compactness_worst            -2.7245     2.4740  -1.101   0.2708
concavity_worst               4.1001     2.0425   2.007   0.0447 *
`\\`concave points_worst\\``  0.5242     2.5825   0.203   0.8391
symmetry_worst                2.9932     1.2657   2.365   0.0180 *
fractal_dimension_worst       1.5024     1.3801   1.089   0.2763
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 602.315  on 455  degrees of freedom
Residual deviance:  60.452  on 434  degrees of freedom
AIC: 104.45

Number of Fisher Scoring iterations: 10
```

Figure 5: Logistic Regression Model Output

	Odds Ratio	p-value
Intercept	0.64	0.23
PC1	0.05	< 0.05
PC2	7.27	< 0.05
PC3	0.44	< 0.05
PC4	0.39	< 0.05
PC5	6.48	< 0.05

	%
AUC	98.90
Best Probability Cutoff	46.00
Error	3.51
PPV	92.11
NPV	98.68
Sensitivity	97.22
Specificity	96.15
Prevalence	31.58

Figure 6: Logistic Regression with PCA (Validation Set Approach)