Breast Cancer Statistical Analysis

Jas Kainth

02/10/2020

Model Selection

Logistical Classifier

The first model we will create is a logistic classifier. We will perform backward elimination, removing covariates based on the goodness of fit, using the AIC statistic, and looking at p-values. The basic form of a logistic model is the following;

$$Y_i \sim \text{Binomial}(\mu_i)$$
$$\ln \frac{\mu_i}{1-\mu_i} = \mathrm{X}_i\beta$$

where Y_i is the i^{th} person. In this model, the $\mathrm{X}_i\beta$ represents the log-odds and the μ_i represents the probability. To get from odds to probability, use probability $= \frac{\text{odds}}{1+\text{odds}}$ or $\mu_i = \frac{\exp \mathrm{X}_i\beta}{1+\exp \mathrm{X}_i\beta}$.

Table 1: Logistic Model 1 Summary Output

term	estimate	std.error	statistic	p.value
Baseline	-10.2562297	1.3530124	-7.5802926	0.0000000
Clump Thickness	0.6997706	0.1867931	3.7462342	0.0001795
Uniformity of Cell Size	-0.1213563	0.2209680	-0.5492029	0.5828662
Uniformity of Cell Shape	0.3657786	0.2440610	1.4987182	0.1339468
Marginal Adhesion	0.3669377	0.1541567	2.3802901	0.0172990
Single Epithelial Cell Size	-0.1232997	0.2000401	-0.6163747	0.5376473
Bare Nuclei	0.3992782	0.1112554	3.5888440	0.0003321
Bland Chromatin	0.4653600	0.1774797	2.6220466	0.0087403
Normal Nucleoli	0.3005112	0.1265104	2.3753873	0.0175305
Mitoses	0.2938408	0.3299858	0.8904652	0.3732161

Table 2: Logistic Model 2 Summary Output

term	estimate	std.error	statistic	p.value
Baseline	-10.0524710	1.2677702	-7.9292534	0.0000000
Clump Thickness	0.6580567	0.1774214	3.7090044	0.0002081
Uniformity of Cell Shape	0.2610548	0.1789558	1.4587671	0.1446292
Marginal Adhesion	0.3314126	0.1508444	2.1970489	0.0280170
Bare Nuclei	0.3815363	0.1035154	3.6857926	0.0002280
Bland Chromatin	0.4179196	0.1691267	2.4710447	0.0134719
Normal Nucleoli	0.2640726	0.1171389	2.2543543	0.0241739
Mitoses	0.2867237	0.3280821	0.8739389	0.3821516

Table 3: Logistic Model 3 Summary Output

term	estimate	std.error	statistic	p.value
Baseline	-10.1189029	1.2871284	-7.861611	0.0000000
Clump Thickness	0.7353629	0.1669259	4.405324	0.0000106
Uniformity of Cell Shape	0.2710022	0.1746600	1.551598	0.1207584
Marginal Adhesion	0.3480460	0.1501143	2.318540	0.0204200
Bare Nuclei	0.3707254	0.1033853	3.585861	0.0003360
Bland Chromatin	0.4225517	0.1680030	2.515144	0.0118984
Normal Nucleoli	0.2669270	0.1173013	2.275568	0.0228719

Table 4: Logistic Model 4 Summary Output

term	estimate	std.error	statistic	p.value
Baseline	-10.4953530	1.3144100	-7.984839	0.0000000
Clump Thickness	0.8258562	0.1631345	5.062426	0.0000004
Marginal Adhesion	0.3972091	0.1476173	2.690804	0.0071280
Bare Nuclei	0.4338568	0.0946070	4.585888	0.0000045
Bland Chromatin	0.4994856	0.1536062	3.251729	0.0011471
Normal Nucleoli	0.3345007	0.1069814	3.126719	0.0017677

It should be noted that the point estimates have not yet been exponentiated. We will take a look at the exponentiated point estimates and their interpretation once we have a final model. Taking a look at Table 1, we see that a lot of the p-values are greater than 0.05. This would suggest that they are not important in predicting the type of cancer but when we were exploring the data in the EDA it seemed that when any of the predictors increased, the cancer was more likely to be malignant. Therefore, the high p-values are probably due to the high correlation among predictors. Also, the AIC of this model is 101.93. AIC is a goodness of fit statistic. By itself, it does not mean anything but when we are comparing models we generally want to pick the model with the lowest AIC.

To get to model 2, we remove 2 predictors with the highest p-value which are 'Uniformity of Cell Size' & 'Single Epithelial Cell Size'. For this model, we again get non-significant p-values. But, we note that the AIC for this model decreases to 98.675 which means this model is better than the previous model. However, let's see if we can do better by removing the predictor with the greatest p-value, which is 'Mitoses'.

For model 3, there is only one predictor with a p-value greater than 0.05, which is 'Uniformity of Cell Shape'. The AIC for this model is 97.766 which is lower than the previous model, so that's a good sign. Let's see if we can remove the non-significant predictor and get a better model.

For model 4, we see that there are no predictors with p-values higher than 0.05. However, the AIC for this model is 98.494 which is higher for this model than the previous model which would suggest that model 3 fits our data better than model 4. Let's explore this further using the likelihood ratio test and using cross-validation.

Table 5: Likelihood Ratio Test of Model 3 vs Model 4

#Df	LogLik	Df	Chisq	Pr(>Chisq)
7	-41.88281	NA	NA	NA
6	-43.24714	-1	2.728667	0.0985615

Table 6: Confusion Matrix of Model 3

	0	1
0	87	4
1	2	44

Table 7: Confusion Matrix of Model 4

	0	1
0	87	4
1	2	44

Table 5 shows the results of the likelihood ratio test. The p-value of this test is greater than 0.05, which would lead us to believe that Model 3 is better than Model 4. Tables 6 & 7 are confusion matrices of Model 3 and 4, respectively. As we can see, they are the same, and both have an accuracy of 0.9562044.

Table 8: Results of Cross-Validation for Model 3

parameter	Accuracy	Kappa	AccuracySD	KappaSD
none	0.9688552	0.9318115	0.0173743	0.0376174

Table 9: Results of Cross-Validation for Model 4

parameter	Accuracy	Kappa	AccuracySD	KappaSD
none	0.9652525	0.923103	0.0201006	0.0447867

We see that the (k-fold) cross-validation results show us accuracy and kappa. The difference between accuracy and kappa is accuracy is the percentage of correctly classified instances out of all the instances. Kappa, however, takes into account the possibility of the correct result occurring by chance. Table 8 & 9 looks at the results of the cross-validation of models 3 & 4. We see that the accuracy and kappa are higher for model 3 and the SD (standard deviation) of accuracy and kappa are lower for model 3. This means model 3 predicts the results more accurately and also the accuracy deviates a less from one test to another.

This is why we will choose model 3, where we predict the cancer type using 'Clump Thickness', 'Uniformity of Cell Shape', 'Marginal Adhesion', 'Bare Nuclei', 'Bland Chromatin' & 'Normal Nucleoli', as the optimal model. We will interpret the results of that model and also create a plot to see how the predictors affect the odds of having malignant cancer. Also, moving forward, when we create models using different classification techniques, we will be using these predictors for those rather than performing backwards elimination again. The reasoning behind this is we don't get a summary output for those models as we do for logistic regression so we would have to completely base our predictors on the accuracy of the test set rather than doing more tests as we performed for the logistic regression.

Table 8 illustrates the final accuracy statistics of our model.

Table 10: Effects on the Odds of Having Malignant Cancer

term	estimate	conf.low	conf.high	p.value
Baseline	0.0000403	0.0000021	0.0003547	0.0000000
Clump Thickness	2.0862389	1.5506523	3.0184372	0.0000106

term	estimate	conf.low	conf.high	p.value
Uniformity of Cell Shape	1.3112779	0.9534260	1.9101739	0.1207584
Marginal Adhesion	1.4162974	1.0744012	1.9479781	0.0204200
Bare Nuclei	1.4487852	1.1899850	1.7957194	0.0003360
Bland Chromatin	1.5258501	1.1131882	2.1686469	0.0118984
Normal Nucleoli	1.3059451	1.0454146	1.6649755	0.0228719

Table 10 shows the final summary output of our model. The coeffcients have been exponentiated here, which means the point estimates represents odds rather than log-odds. The 'Baseline' represents the odds of having malignant cancer given the remainded of the predictors have a value of 1. The probability is then $\frac{0.0000403}{1+0.0000403} \approx 0.0000403 = 4.03 \times 10^{-5}$. For the remainder of the point estimates, the are multiplicative factors. This means, looking at 'Clump Thickiness' for example, if we increase the value of the predictor by 1, all else the same, the odds would increase by 108.6% (2.086 - 1 = 1.086 or 109%). Then, to get to probability from odds, we use probability $= \frac{\text{odds}}{1+\text{odds}}$.

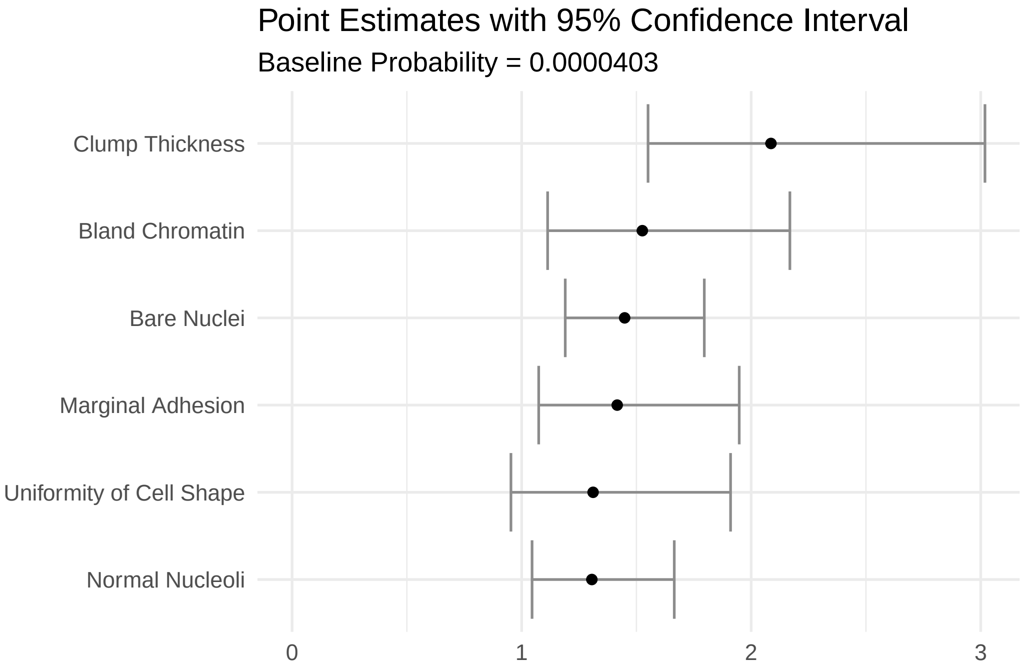

Figure 1: Effects by Predictors to the Odds of Having Malignant Breast Cancer

Figure 1 is a visualization that shows how predictors affect the odds of having malignant cancer. We note that only ‘Uniformity of Cell Shape’ has an error bar that crosses 1. This error bar is a 95% confidence interval and we expected this since the p-value is greater than 0.05. It should be noted that it crosses 1, not 0, since the terms have been exponentiated ($\exp(0) = 1$). Based on the point estimates, it seems that ‘Clump Thickness’ has the largest effect on the odds, however, the error bar is quite large. The effects from ‘Bare Nuclei’ and ‘Normal Nucleoli’ aren’t nearly as large but the error bars are more narrow for these estimates. But, we do note that all of them have point estimates greater than 1, which means that increasing the values of these predictors results in an increased odds (and therefore probability) of having malignant cancer.

K-th Nearest Neighbor (KNN)

The next model which we will create is a k-th nearest neighbor (KNN) model. This model checks the k nearest points (neighbors) to the new point and then places it in the group that has more neighbors from the corresponding group. For this model (and all future models), we will use the same covariates as the 3rd logistic model (Clump Thickness, Marginal Adhesion, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Uniformity of Cell Shape). First, we will perform cross validation on the model, using these predictors, to pick the optimal k (the number of neighbors the model takes into consideration).

Table 11: Result of cross-validation for knn model

k	Accuracy	Kappa	AccuracySD	KappaSD
1	0.9524242	0.8953108	0.0286778	0.0619196
2	0.9651852	0.9237776	0.0327257	0.0713646
3	0.9671044	0.9282512	0.0340702	0.0743881
4	0.9743771	0.9442909	0.0246006	0.0532361
5	0.9725589	0.9403198	0.0231240	0.0500016
6	0.9688889	0.9323388	0.0244268	0.0528260
7	0.9707071	0.9362136	0.0246679	0.0533643
8	0.9707071	0.9361210	0.0300392	0.0655363
9	0.9725253	0.9400921	0.0313098	0.0683080
10	0.9725253	0.9400921	0.0313098	0.0683080
11	0.9743434	0.9438658	0.0275147	0.0603795
12	0.9725253	0.9397208	0.0301138	0.0663159
13	0.9725253	0.9397208	0.0301138	0.0663159
14	0.9707071	0.9354840	0.0335073	0.0743080
15	0.9688889	0.9314129	0.0322093	0.0714406

Table 11 shows the results of the cross-validation. We want to pick the k which corresponds to the highest accuracy. In case there was a tie, we would pick the larger value of k, but that is not the case for this model. The optimal k for this model is 4 (which also happens to have the largest value for kappa). As we can see, the accuracy of this model is 0.9743771. For comparison, the accuracy for the optimal logistic model was 0.9688552 which means this model performs slightly better than the logistic model.

Table 12: Confusion Matrix for KNN model

	0	1
0	86	2
1	3	46

Table 12 is the confusion matrix of this model, comparing the predicted results to the actual results from the test set. We cannot see the effect of the covariates on the response to this model. Therefore, we cannot examine if increasing the value of covariates increases or decreases the probability of malignant cancer. So, when comparing this model to the logistic regression, this model gives slightly more accurate results but the other model has more interpretable results.

Support Vector Machine (SVM)

The next model which we will create is a support vector machine (SVM). This model uses a line, plane or hyperplane (depending on the number of dimensions) to differentiate the two groups. The shape of the hyperplane can take many shapes, through the parameter of the kernel, so first, we will find the kernel which has the highest accuracy and further tune that model.

Table 13: Confusion matrix for SVM model using a linear kernel

	0	1
0	86	3
1	3	45

Table 14: Confusion matrix for SVM model using a polynomial kernel

	0	1
0	88	6
1	1	42

Table 15: Confusion matrix for SVM model using a radial kernel

	0	1
0	85	2
1	4	46

Table 16: Confusion matrix for SVM model using a sigmoid kernel

	0	1
0	85	1
1	4	47

Using the confusion matrices, we find that the accuracy of the SVM model is the following

Table 17: Accuracy of SVM under different kernels

Kernel	Accuracy
Linear	0.9562044
Polynomial	0.9489051
Radial	0.9562044
Sigmoid	0.9635036

Table 17 shows the highest accuracy is with the sigmoid kernel, however, the train function we use the apply cross-validation doesn't have a method for the sigmoid kernel so we will use the next highest accuracy (which is linear and radial).

Table 18: Results from cross-validation with radial kernel

sigma	C	Accuracy	Kappa	AccuracySD	KappaSD
1.022693	0.25	0.9507071	0.8965516	0.0437129	0.0899571
1.022693	0.50	0.9671380	0.9299289	0.0318150	0.0669896
1.022693	1.00	0.9689226	0.9331312	0.0271855	0.0579107

Table 19: Results from cross-validation with linear kernel

C	Accuracy	Kappa	AccuracySD	KappaSD
1	0.9725241	0.9397818	0.0154981	0.0340788

With the radial kernel, the C represents the cost of constraints violation, which is what we are attempting to maximize the accuracy with respect to. As we can see, the highest accuracy is with C = 1 (which is the default with the svm function which is what we used to create the model initially so we don't need to tune that model). With the linear kernel, we note that the accuracy is higher than the accuracy of the radial kernel. Again, with this model, we cannot directly see the effects of the predictors as we could with the logistic regression classifier.

Random Forest Classifier

The next model which we will use is the random forest classifier. A random forest classifier is a type of ensemble learning algorithm, which is an algorithm where you take multiple machine learning algorithms and put them together to get another machine learning algorithm. In this case, we are taking multiple decision trees and putting them together. For the model, we use the default of 500 as the number of (decision) trees which is a relatively large number. When the number of trees increases beyond a point the performance plateaus, which means the efficiency almost peaks.

Table 20: Confusion Matrix for Random Forest Regression

	0	1
0	87	2
1	2	46

Table 20 shows the confusion matrix of our random forest regression model. As we see, our model got 132 correct results out of 137 (an accuracy of 0.9708029). However, one parameter which we can tune is the number of variables the function tries at each split. By default, this value is 2, so let's use cross-validation to see if we can increase the efficiency of our model.

Table 21: Results of cross-validation for the random forest model

mtry	Accuracy	Kappa	AccuracySD	KappaSD
2	0.9743434	0.9439497	0.0215224	0.0468592
4	0.9706734	0.9357884	0.0262561	0.0575193

mtry	Accuracy	Kappa	AccuracySD	KappaSD
6	0.9707071	0.9359223	0.0214844	0.0468037

Table 21 shows the results of the cross-validation where mtry is the parameter that we are tuning. As we see the highest accuracy is achieved with mtry = 2, which was the default value.

Naive Bayes Classifier

For the naive bayes classifier, we use bayes theorem

$$\begin{aligned} P(\text{Malignant Cancer} \mid \text{X}) &= \frac{P(\text{X} \mid \text{Malignant Cancer}) \cdot P(\text{Malignant Cancer})}{P(\text{X})} \\ \text{Posterior Probability} &= \frac{\text{Likelihood} \cdot \text{Prior Probability}}{\text{Marginal Likelihood}} \end{aligned}$$

where X is our data (i.e. the matrix of predictors).

Table 22: Confusion matrix of the naive bayes classifier

	0	1
0	85	1
1	4	47

Table 22 is the confusion matrix of naive bayes classifier. Our model gets 132 correct predictions out of 137 (accuracy of 0.9635036). This is pretty good, let's try tuning some parameters to get the best accuracy we can

Table 23: Results of cross-validation for the naive bayes model

fL	usekernel	adjust	Accuracy	Kappa	AccuracySD	KappaSD
1.0	TRUE	1.0	0.9743434	0.9442379	0.0213944	0.0462635
1.0	TRUE	1.5	0.9743434	0.9442379	0.0213944	0.0462635
1.0	TRUE	2.0	0.9761279	0.9481726	0.0244851	0.0529450
1.0	TRUE	2.5	0.9724916	0.9400305	0.0215728	0.0464766
1.5	TRUE	1.0	0.9743434	0.9442379	0.0213944	0.0462635
1.5	TRUE	1.5	0.9743434	0.9442379	0.0213944	0.0462635
1.5	TRUE	2.0	0.9761279	0.9481726	0.0244851	0.0529450
1.5	TRUE	2.5	0.9724916	0.9400305	0.0215728	0.0464766
2.0	TRUE	1.0	0.9743434	0.9442379	0.0213944	0.0462635
2.0	TRUE	1.5	0.9743434	0.9442379	0.0213944	0.0462635
2.0	TRUE	2.0	0.9761279	0.9481726	0.0244851	0.0529450
2.0	TRUE	2.5	0.9724916	0.9400305	0.0215728	0.0464766
2.5	TRUE	1.0	0.9743434	0.9442379	0.0213944	0.0462635
2.5	TRUE	1.5	0.9743434	0.9442379	0.0213944	0.0462635
2.5	TRUE	2.0	0.9761279	0.9481726	0.0244851	0.0529450
2.5	TRUE	2.5	0.9724916	0.9400305	0.0215728	0.0464766

Tuning our parameters, we see that the accuracy peaks when fL and adjust = 2. The accuracy for our model is 0.9761279 along with a standard deviation of the accuracy of 0.0232131.

Summary

Table 24: Summary Statistics of Models Presented

Model	Accuracy	Kappa	AccuracySD	KappaSD	Notes
Logistic	0.9688552	0.9318115	0.0173743	0.0376174	
KNN	0.9743771	0.9442909	0.0246006	0.0532361	K = 4
SVM	0.9725241	0.9397818	0.0154981	0.0340788	Linear Kernel
Random Forest	0.9743434	0.9439497	0.0215224	0.0468592	Variables at Split = 2
Naive Bayes	0.9761279	0.9481726	0.0244851	0.0529450	fL & adjust = 2

Table 24 summarizes all of our models, in the order they were presented where we pick the model with the highest accuracy from each model type. We see the best model in terms of accuracy is the Naive Bayes model. But it should be noted that all of the accuracies are quite similar. The percent difference between the highest accuracy and lowest accuracy is 0.747842%. However, the standard deviation of the logistic and SVM classifiers are much lower than the others (especially the SVM). This means that their accuracies deviate less when compared to the other models. For comparison, the percent difference between our most accurate model (Naive Bayes) and our model with the lowest standard deviation (SVM) is 44.95388%! This means that when we pick our "best" model, there are many factors to consider.

The most important factor to consider is interpretability. This, of course, is only true when our interpretable models perform well which in this case they do. Only with the logistic model can we see how the predictors affect the odds (or probability) of having malignant cancer. The other models may predict the results better than this model but we can interpret the results and understand the results a lot better with this model (see Figure 1 & Table 10). Therefore, if we just want to predict the results when given information, we might choose the SVM model or the naive Bayes, but if we want to report on our findings, the logistic classifier is definitely the most optimal model.