Non-invasive Prediction Models for Malignant Breast Tumor

Olivia Fan, Alicia Gong

18 December, 2022

Introduction

Breast cancer is the most common cancer worldwide and the most common cancer diagnosed in the US (Mayo). Each year in the US, about 264,000 cases of breast cancer are diagnosed in women and about 2,400 in men (CDC).

Early diagnosis of the condition is crucial to improve the survival rate and relieve suffering in patients. Mammography is an effective X-ray imaging technology that detects breast cancer early. Classically, benign or malignant breast tumors are diagnosed by radiologists' interpretation of mammograms based on clinical parameters. However, diagnosing cancer is challenging even for the most skilled doctors. Since masses are heterogeneous, clinical parameters supply limited information on mammography mass. The symptoms are often shared with diseases and conditions that are unrelated to cancer, leading doctors to improperly diagnose the disease.

Cancerous lumps are often confused for blocked milk ducts, breast cysts, and other benign conditions. According to an expansive study conducted by Dartmouth College, the University of Vermont, and the Fred Hutchinson Cancer Research Center, and published in the March 2015 issue of the Journal of American Medical Association, approximately 13% of the diagnoses missed Stage 1 breast cancer. Meanwhile, 48% failed to detect atypia hyperplasia, a precursor to breast cancer. A significant number also over-diagnosed atypia hyperplasia.

There is, therefore, an urgent need to find new tools that can identify patients with breast cancer. Our study aims to build supervised machine-learning models to predict the diagnosis of breast cancer and understand the most important variables, to assist doctors and radiologists in accurately interpreting mammography imaging.

We built 3 models in total: Lasso penalized logistic regression, SVM, and Random Forest.

Data

We obtain the Breast Cancer Wisconsin (Diagnosis) Data Set from Kaggle. The dataset contains diagnosis results and features of the cell nuclei computed from a digitized image of a fine needle aspirate (FNA) of a breast mass for 568 patients. The size of the nucleus is expressed by the features radius and area. The shape is expressed by the features smoothness, concavity, compactness, concave points, symmetry, and fractal dimension. The perimeter expresses both the size and shape of the nucleus. A higher value of shape features corresponds to a less regular contour and, therefore, to a higher probability of malignancy. For each of the features the mean value, worst value (mean of the three largest values), and standard error are computed for each image, resulting in 30 features of 568 images.

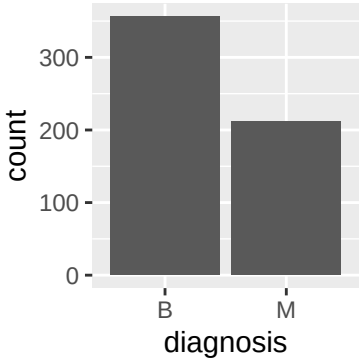
Data Processing

The original dataset contains a blank column '. . . 33,' so we dropped it. We also dropped the 'id' column, and rename several columns that contains blank space in their names.

In order to fit SVM on the data, we encode the `diagnosis` variable into a factor variable with level 1 and -1:

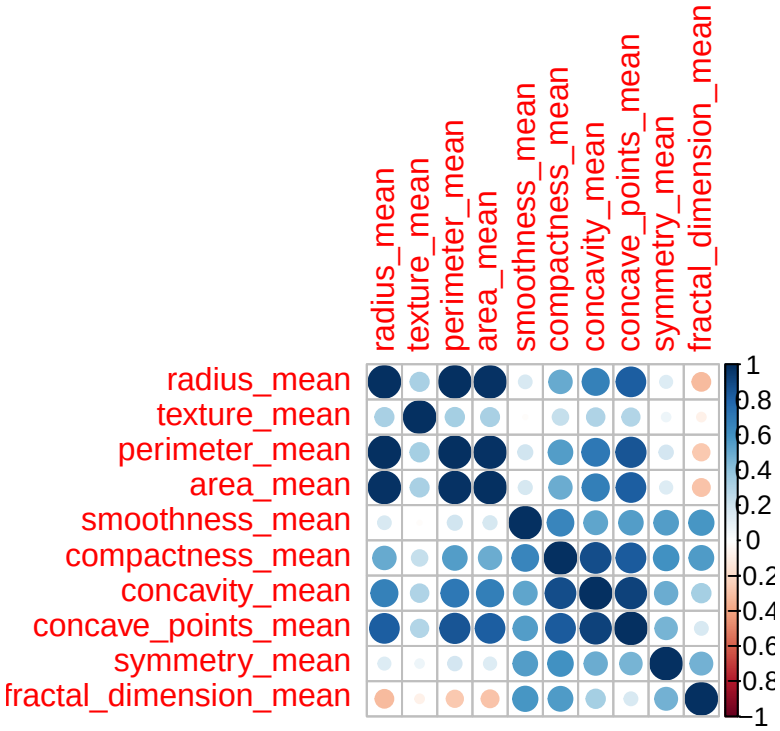
We partition the data into training and testing sets using a 70-30 percentage split(70% of the original data as the training set, and 30% as the testing set):

Exploratory Data Analysis



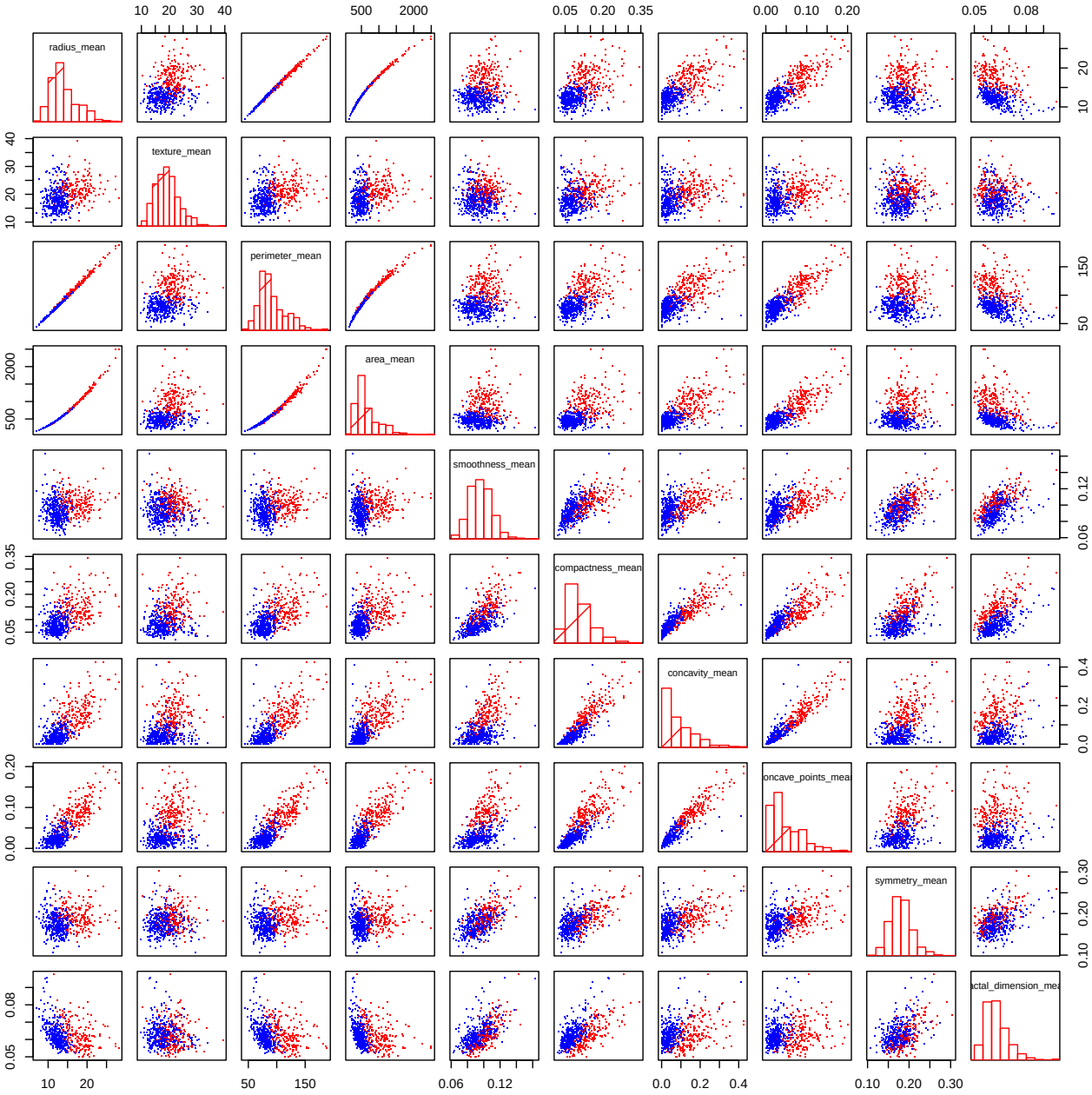
The bar plot shows that there is a larger number of benign than malignant cancer.

We divide the data into 3 categories according to their features.



Major observations:

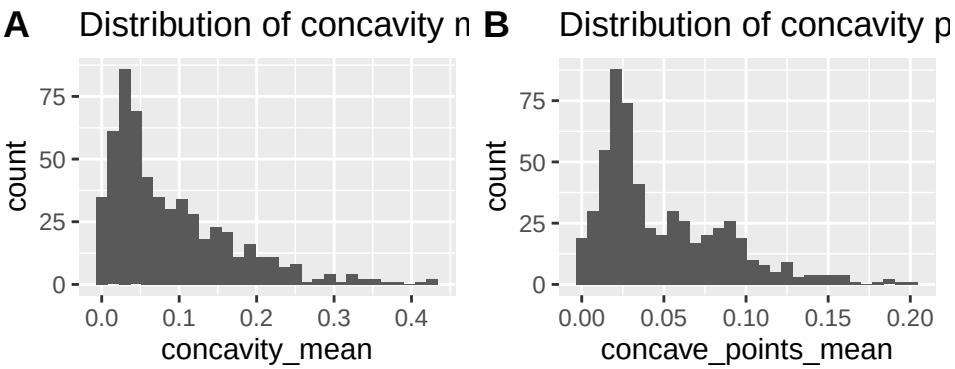
- Radius_mean, perimeter_mean, and area_mean are highly correlated.
- Compactness_mean, concavity_mean and concave_points_mean are highly correlated.



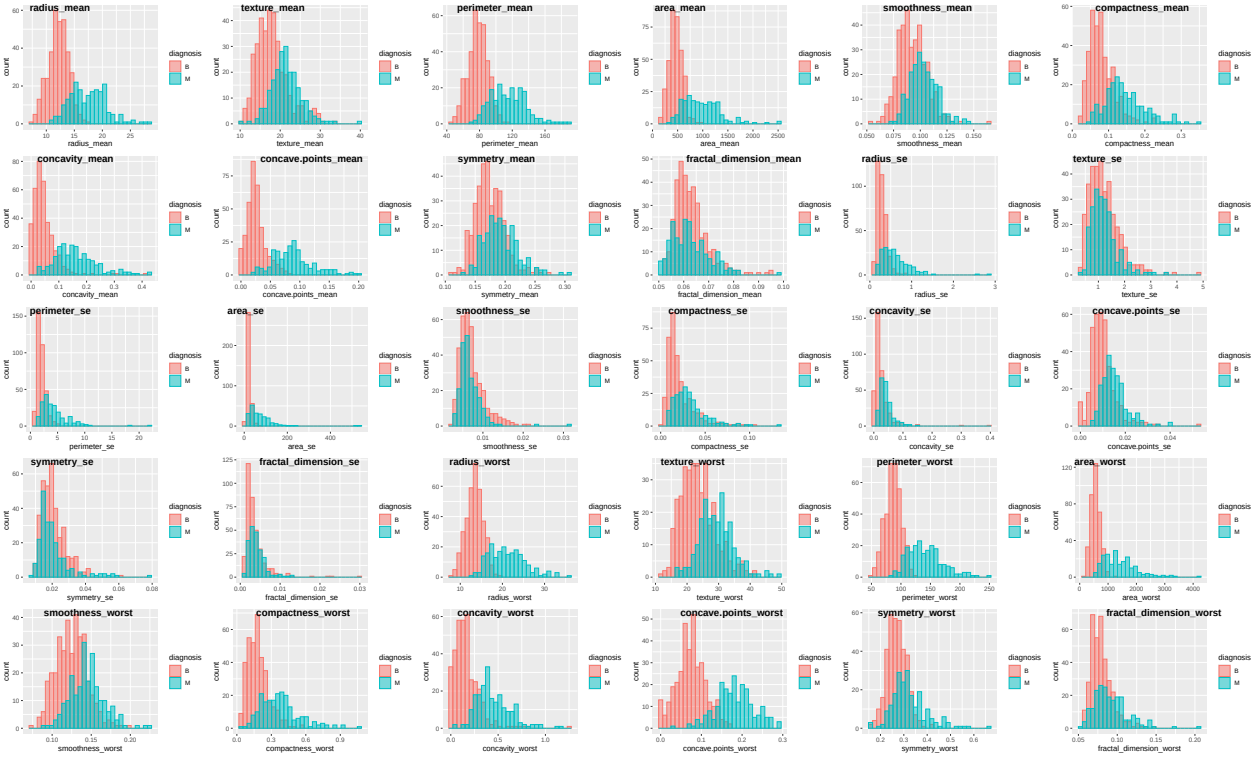
We observe from the pairwise scatterplot matrix above that the two classifications seem to be generally separable, with distinct regions in the visualization that cleanly cluster without much mingling or mixing. Overall across malignant and benign tumors, there seems to be a strong positive linear relationship between `radius_mean` and `parameter_mean`, `radius_mean` and `area_mean`, as well as `area_mean` and `parameter_mean`, which hints again at the collinearity issue which we will later tackle at through variable selection. While the two classifications together constitute a roughly linear relationship between predictors, malignant tumors (red) generally associate with higher values in both predictors accumulating in the right top corner, while benign tumors (blue) generally associate with lower ones in the left bottom corner.

We observe that there does not seem to be a separating hyperplane for the two classes for predictors in the relationship between `texture_mean` and `symmetry_mean`, and between `smoothness_mean` and `fractal_dimension_mean` since the observations in two classes mingle together. This hints at the fact that these predictors might not be helpful for the two class classification problem, which we will filter out in our model through variable selection.

The distributions of the predictors seem to be all unimodal, with no apparent outliers and generally right-skewed, with `concavity_mean` and `concave_points_mean` being particularly right-skewed, hinting at the high correlation between the two predictors. Hence we want to consider including only one of them in our model. We take a closer look at the distributions of these two predictors here:



After deriving the histogram comparing distributions of predictors based on the two classifications, we would like to find features with little overlap between benign and malignant classes which will likely to be significant for diagnosis. We plot the distribution of the predictors separated by the benigh and malignant classes, and observe again that predictors associated with texture, smoothness, symmetry and fractal dimension are inseparable and therefore might not be helpful for the classification problem. For example, the distributions of `smoothness_se` for benign and malignant cancers almost completely overlap, same do `symmetry_se`, `fractal_dimension_se` and `texture_se`. Therefore, we will consider filter out these predictors in our final model.



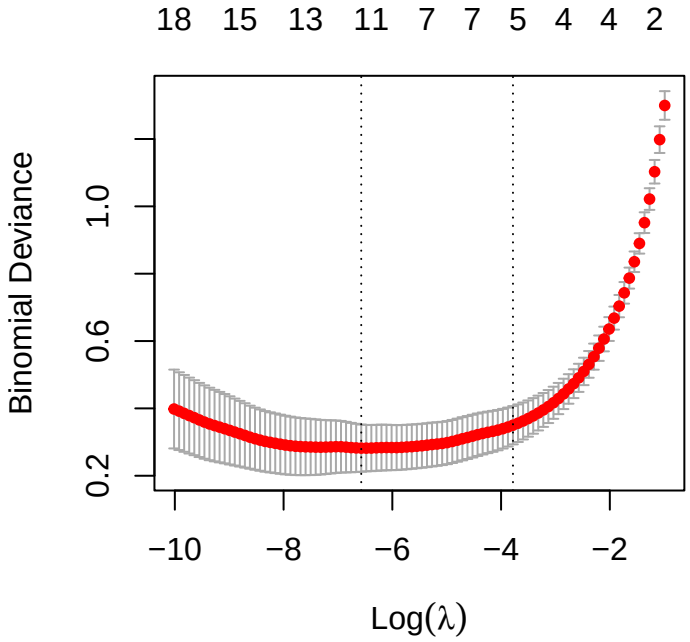
Methodology

Model 1: LASSO-Penalized Logistic Regression

We decided to use a LASSO-penalized logistic regression model to perform variable selection by gauging insights into which predictors are the most contributive, since less significant variables are forced to be exactly zero, and the most significant variables are kept in the final model. As explained in our exploratory data analysis above, we filter out predictors associated with texture, smoothness, fractal dimension and symmetry in our model due to the lack of separation in the values of these predictors for the benign and malignant tumor classes.

Hyperparameter Tuning

We fitted the LASSO-penalized logistic regression model using the optimal hyperparameter $\lambda = 0.0026830$ via cross validation. To explore the interaction between symmetry and the mean for number of concave portions of the contour, we included the interaction term `symmetry_worst*concave_points_mean`.




```
## [1] 0.001398917
```

	coefficient
(Intercept)	-10.4871
radius_mean	-0.1130
compactness_mean	-33.6480
concavity_mean	14.3918
concave_points_mean	19.0997
area_se	0.0944
compactness_se	-30.3992
concavity_se	-31.2549
concave_points_se	34.8557
radius_worst	0.0112
area_worst	0.0049
compactness_worst	4.9213
concavity_worst	7.6360
concave_points_worst	26.8649
compactness_mean:concave_points_mean	74.5208

Logistic Model

Model Interpretation

Prediction

Using the logistic regression model, besides classification we also want to understand uncertainty - more specifically, predictive probabilities that a tumor is benign or malignant given the values of the predictors:

texture_mean	concave_points_mean	radius_se	texture_se	area_se	smoothness_se	compactness_se	fractal_dimension_se	radius_worst	texture_worst	smoothness_worst	concavity_worst	concave_points_worst	symmetry_worst	probabilities	predicted_class
10.38	0.147	1.095	0.905	153.40	0.006	0.049	0.006	25.38	17.33	0.162	0.712	0.265	0.460	1.000	M
17.77	0.070	0.543	0.734	74.08	0.005	0.013	0.004	24.99	23.41	0.124	0.242	0.186	0.275	1.000	M
21.82	0.094	0.306	1.002	24.32	0.006	0.035	0.004	15.49	30.73	0.170	0.539	0.206	0.438	0.982	M
22.61	0.080	0.212	1.169	19.21	0.006	0.059	0.008	15.03	32.01	0.165	0.694	0.221	0.360	0.976	M
20.13	0.053	0.473	1.240	45.40	0.006	0.012	0.002	19.07	30.88	0.146	0.291	0.161	0.303	0.991	M
20.68	0.103	0.569	1.073	54.18	0.007	0.025	0.004	20.96	31.48	0.179	0.478	0.207	0.371	1.000	M
22.15	0.095	0.758	1.017	112.40	0.006	0.019	0.002	27.32	30.88	0.151	0.537	0.239	0.277	1.000	M
15.71	0.031	0.185	0.748	14.67	0.004	0.019	0.002	14.50	20.49	0.131	0.189	0.073	0.318	0.002	B
20.25	0.077	0.853	1.849	93.54	0.011	0.027	0.004	21.31	27.26	0.134	0.345	0.149	0.234	1.000	M
18.70	0.052	0.482	1.030	41.00	0.006	0.034	0.006	16.82	28.12	0.164	0.696	0.155	0.476	0.994	M

```
## [1] 0.9883041
```

We achieved a prediction accuracy of 0.9883. To interpret the predictions, we see that a patient with tumor with `texture_mean` of 10.38, `concave_points_mean` of 0.147, `radius_se` of 1.095, `texture_se` of 0.905, `area_se` of 153.400, `smoothness_se` of 0.006, `compactness_se` of 0.049, `fractal_dimension_se` of 0.006, `radius_worst` of 25.380, `texture_worst` of 17.33, `smoothness_worst` of 0.162, `concavity_worst` of 0.712, `concave_points_worst` of 0.265, `symmetry_worst` of 0.460 is expected to have a 100% of being diagnosed as malignant tumor. Whereas... is expected to have a 0% of being diagnosed as malignant tumor. Whereas... is expected to have a 87.4% of being diagnosed as malignant tumor.

Model 2: SVM

Linear Kernel SVM

We use the predictors selected by the LASSO penalized logistic regression as predictors for the support vector machine model:

If two predictors have high correlation, we only use one of them:

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##   0.1
##
## - best performance: 0.05301282
##
## - Detailed performance results:
##    cost      error dispersion
## 1 1e-03 0.10333333 0.05485072
## 2 1e-02 0.06307692 0.03823795
## 3 1e-01 0.05301282 0.03872305
```

```
## 4 1e+00 0.05557692 0.02921816
## 5 5e+00 0.05301282 0.02798909
## 6 1e+01 0.05814103 0.02723341
## 7 1e+02 0.06076923 0.03671087
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = diagnosis_binary ~ concave_points_mean +
##     area_se + compactness_se + radius_worst + concavity_worst + concave_points_worst +
##     compactness_mean + compactness_se + concavity_se + concavity_mean +
##     radius_mean + concave_points_se + area_worst + compactness_worst +
##     compactness_mean * concave_points_mean, data = cancer_train,
##     ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)), kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##
## Number of Support Vectors:  74
##
##  ( 36 38 )
##
##
## Number of Classes:  2
##
## Levels:
##  -1 1
```

predict/truth	-1	1
-1	97	4
1	1	69

```
## [1] 0.02923977
```

The misclassification rate is 0.02924.

Radial Kernel SVM

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##     1   0.5
##
## - best performance: 0.07044872
##
## - Detailed performance results:
##     cost gamma      error dispersion
## 1  1e-01   0.5 0.09307692 0.03545438
## 2  1e+00   0.5 0.07044872 0.02835600
```

```
## 3  1e+01   0.5 0.07057692 0.04092698
## 4  1e+02   0.5 0.07307692 0.04197726
## 5  1e+03   0.5 0.07307692 0.04197726
## 6  1e-01   1.0 0.35012821 0.09475806
## 7  1e+00   1.0 0.09064103 0.02936491
## 8  1e+01   1.0 0.09820513 0.03438883
## 9  1e+02   1.0 0.10320513 0.03811214
## 10 1e+03   1.0 0.10320513 0.03811214
## 11 1e-01   2.0 0.35012821 0.09475806
## 12 1e+00   2.0 0.11076923 0.03351610
## 13 1e+01   2.0 0.11326923 0.03746702
## 14 1e+02   2.0 0.11326923 0.03746702
## 15 1e+03   2.0 0.11326923 0.03746702
## 16 1e-01   3.0 0.35012821 0.09475806
## 17 1e+00   3.0 0.18141026 0.12507978
## 18 1e+01   3.0 0.17128205 0.12520558
## 19 1e+02   3.0 0.17128205 0.12520558
## 20 1e+03   3.0 0.17128205 0.12520558
## 21 1e-01   4.0 0.35012821 0.09475806
## 22 1e+00   4.0 0.32737179 0.10661475
## 23 1e+01   4.0 0.31230769 0.11398118
## 24 1e+02   4.0 0.31230769 0.11398118
## 25 1e+03   4.0 0.31230769 0.11398118
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = diagnosis_binary ~ concave_points_mean +
##     area_se + compactness_se + radius_worst + concavity_worst + concave_points_worst +
##     compactness_mean + compactness_se + concavity_se + concavity_mean +
##     radius_mean + concave_points_se + area_worst + compactness_worst +
##     compactness_mean * concave_points_mean, data = cancer_train,
##     ranges = list(cost = c(0.1, 1, 10, 100, 1000), gamma = c(0.5,
##         1, 2, 3, 4)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  186
##
##  ( 106 80 )
##
##
## Number of Classes:  2
##
## Levels:
##  -1 1
```

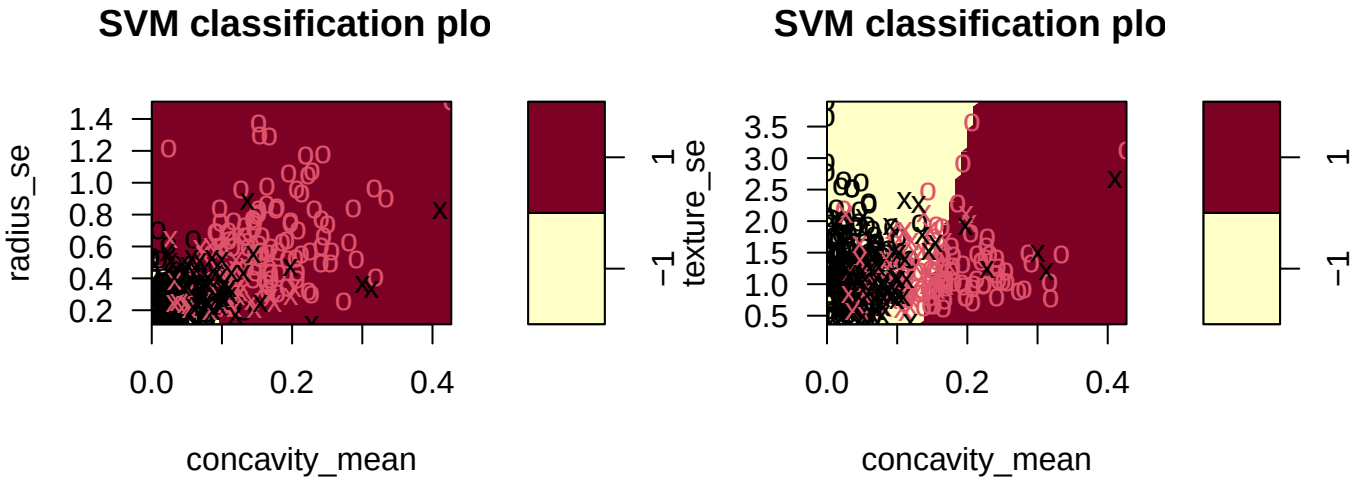
predict/truth	-1	1
-1	96	3
1	2	70

```
## [1] 0.02923977
```

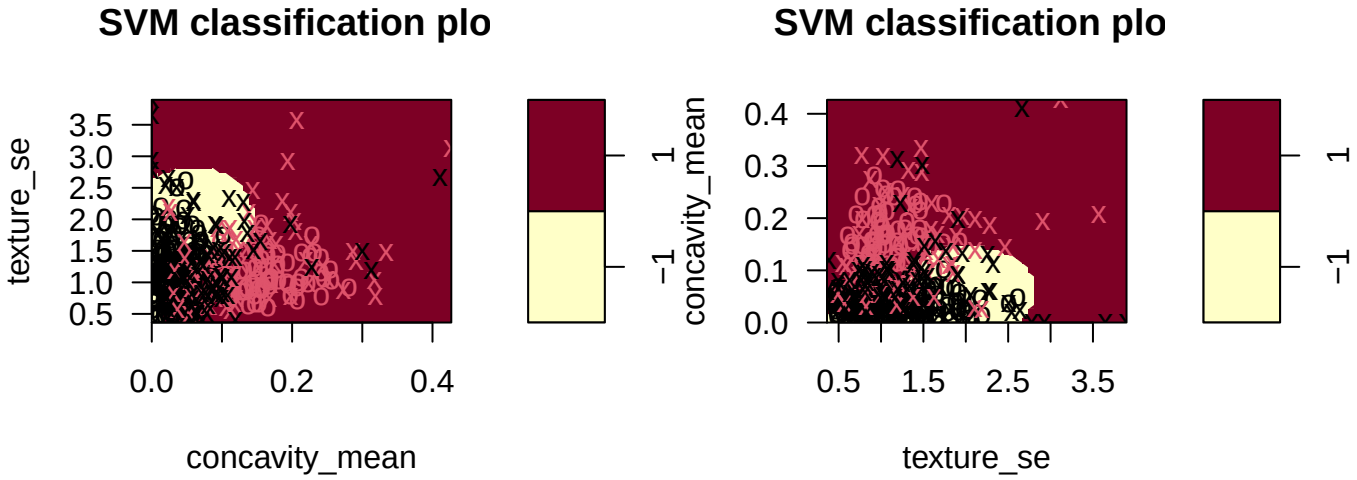
The misclassification rate is 0.02924, which is similar to that of the linear kernel which suggests that the two classes are likely to be linearly separable so that we can find a separating hyperplane using the linear kernel.

SVM Visualization

Linear



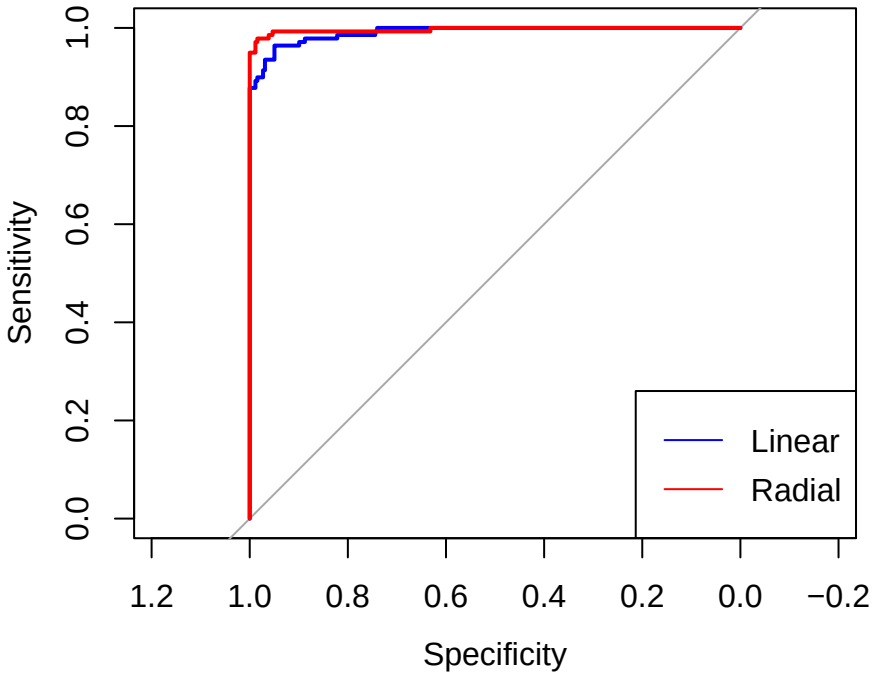
Radial



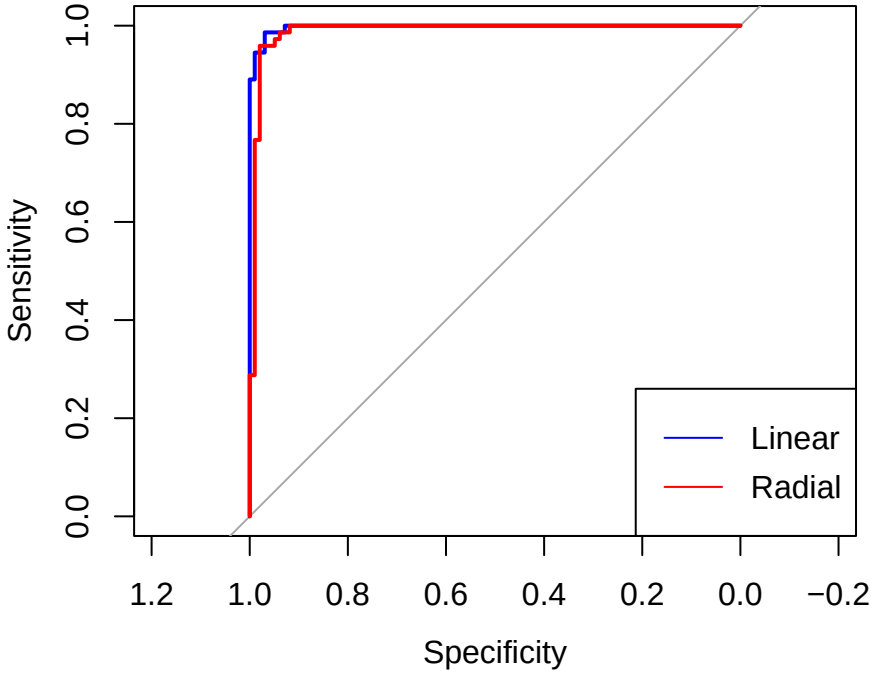
ROC (Linear SVM)

We visualize the ROC curves:

On train data:



On test data:



Even though the radial kernel fits the training data more closely due to its higher complexity, the linear kernel performs better on the test data (since the data is likely to be linearly separable as explained above), we decided to select the model with the linear kernel.

Model 3: Random Forest

For Random Forest, we decided to use group 'Mean' and group 'Worst' separately. This is because, we want to understand the potential difference in prediction given the different level of severity of the patients' conditions. As we have mentioned in the introduction, 'Worst' measures mean of the three largest values. By

building a model with only 'Mean' and 'Worst' variables, we think we can better understand the extreme cases.

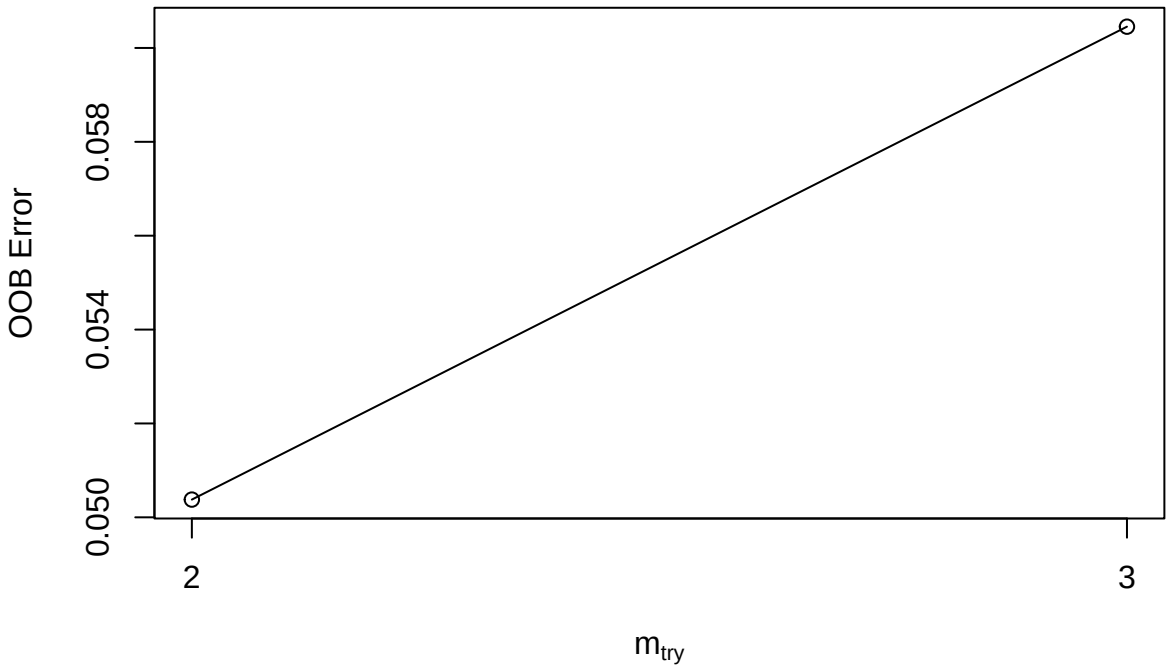
First, We select variables by running a preliminary random forest model with all the variables, to rank their importance.

	MeanDecreaseGini
radius_mean	5.0070158
texture_mean	3.1072100
perimeter_mean	7.5925204
area_mean	9.2068964
smoothness_mean	1.3670226
compactness_mean	1.4568072
concavity_mean	12.6313700
concave_points_mean	18.2770412
symmetry_mean	0.7370356
fractal_dimension_mean	0.9026707
radius_se	2.4673699
texture_se	0.7408013
perimeter_se	3.0568472
area_se	7.1559654
smoothness_se	1.0192210
compactness_se	1.0399584
concavity_se	1.1397170
concave_points_se	1.0101496
symmetry_se	0.9229467
fractal_dimension_se	1.1310419
radius_worst	21.7267522
texture_worst	4.0267581
perimeter_worst	18.8197812
area_worst	15.9260537
smoothness_worst	2.6418060
compactness_worst	3.1437736
concavity_worst	7.1555395
concave_points_worst	23.6956861
symmetry_worst	2.0017696
fractal_dimension_worst	1.1758773

After testing different variables based on their important, and taking into accounts the collinearity issue we discussed in the EDA section, we decided to select the following variables as the predictors: concave_points_worst, area_worst, perimeter_worst, radius_worst, concave_points_mean, perimeter_mean, concavity_worst, area_se.

We chose mtry=2, because after tuning mtry, we found that mtry=2 has the lowest OOB error.

```
## mtry = 2  OOB error = 5.04%
## Searching left ...
## Searching right ...
## mtry = 3     OOB error = 6.05%
## -0.2 0.01
```



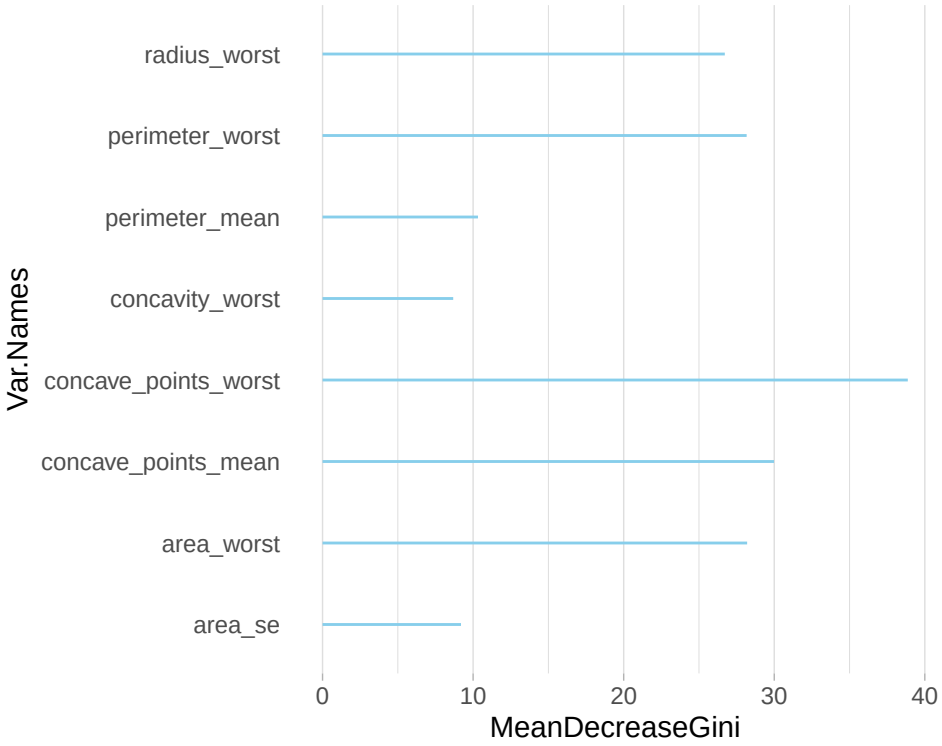
We chose the number of tree to be 500. The number of trees should be chosen carefully, since a high performance of the individual models might lead to overfitting when the number of trees is very high. However, taking 50 trees caused a lower accuracy than taking 500 trees. Therefore, this higher number of trees is chosen.

```
##
## Call:
##  randomForest(formula = diagnosis_binary ~ concave_points_worst +      area_worst + perimeter_worst +
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 5.04%
## Confusion matrix:
##     -1   1 class.error
## -1 251   7  0.02713178
## 1   13 126  0.09352518
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction -1  1
##         -1 96  5
##         1   2 68
##
##                Accuracy : 0.9591
##                  95% CI : (0.9175, 0.9834)
##     No Information Rate : 0.5731
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9159
##
##  Mcnemar's Test P-Value : 0.4497
```

```
##
##             Sensitivity : 0.9796
##             Specificity : 0.9315
##          Pos Pred Value : 0.9505
##          Neg Pred Value : 0.9714
##              Prevalence : 0.5731
##          Detection Rate : 0.5614
##    Detection Prevalence : 0.5906
##       Balanced Accuracy : 0.9555
##
##        'Positive' Class : -1
##
```

The RF model yields a 96% accuracy for the testing set.



From the plot we can see that the most important predictors are: concave_points_worst, concave_points_mean, area_worst, perimeter_worst, and radius_worst. Interestingly, perimeter_worst has high gini coefficient, but perimeter_mean ranks second from the last.

Conclusion & Future Work

Concavity is the severity of concave portions of the contour. A high concavity means that the boundary of the cell nucleus has indentations, and thus is rather rough than smooth. Concave points id the number of concave portions of the contour of the cell nucleus.

Citations

Appendix