# Machine learning classification and clustering for tumor diagnosis

Hanbo Sun, Yue Wang, Tuo Wang

4/15/2017


- 1 Introduction
  - 1.1 Data
  - 1.2 Methods
- 2 Visualizations
  - 2.1 Dimension Reduction
  - 2.2 Clustering
  - 2.3 Variable Selection
  - 2.4 Classification
  - 2.5 SVM and Random Forest on different versions of data
  - 2.6 XGBoost
- 3 Results
  - 3.1 Sensitivity and Recall
  - 3.2 ROC curve of four classifiers on three data version
- 4 Conclusion

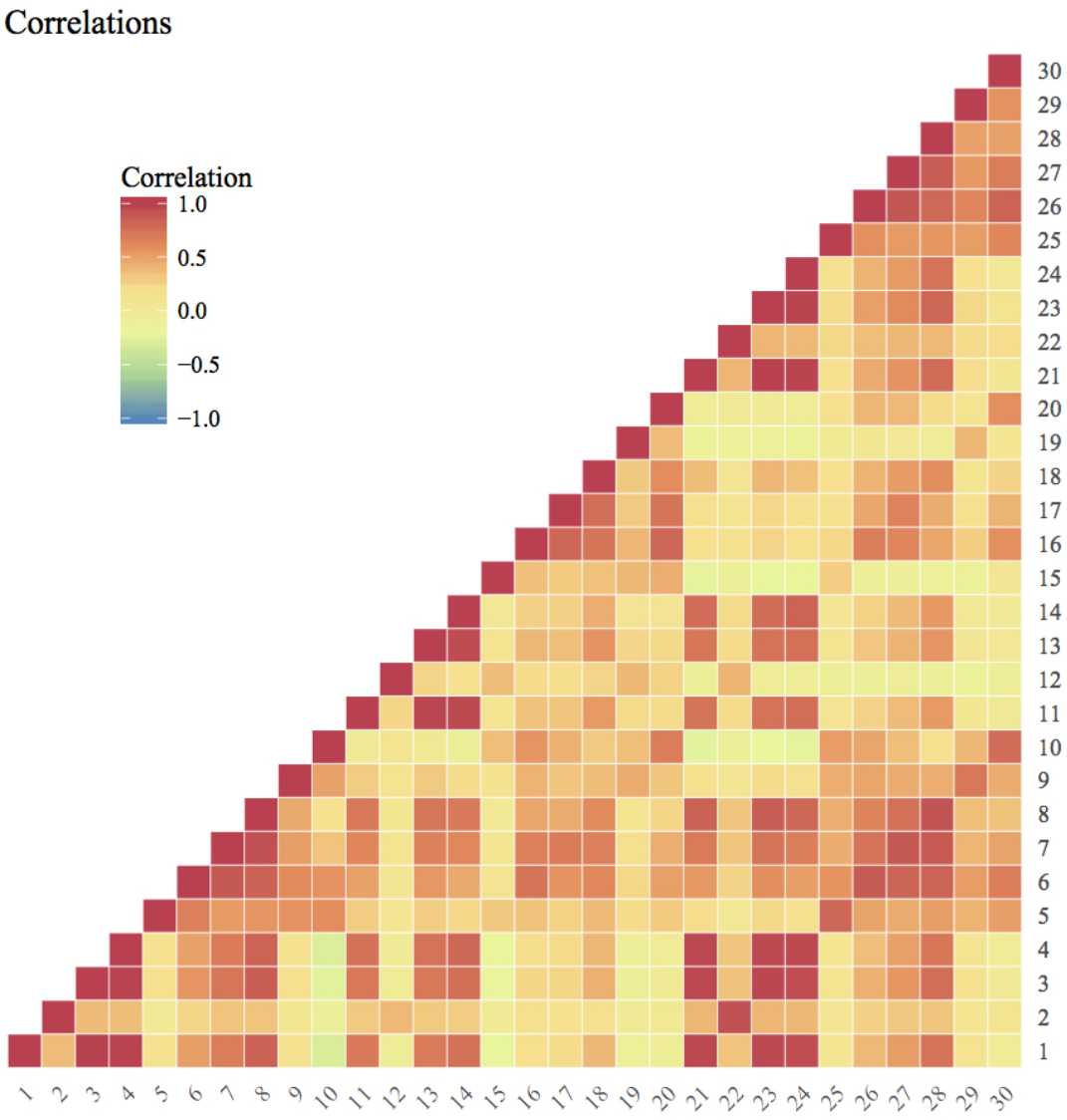

# 1 Introduction

## 1.1 Data

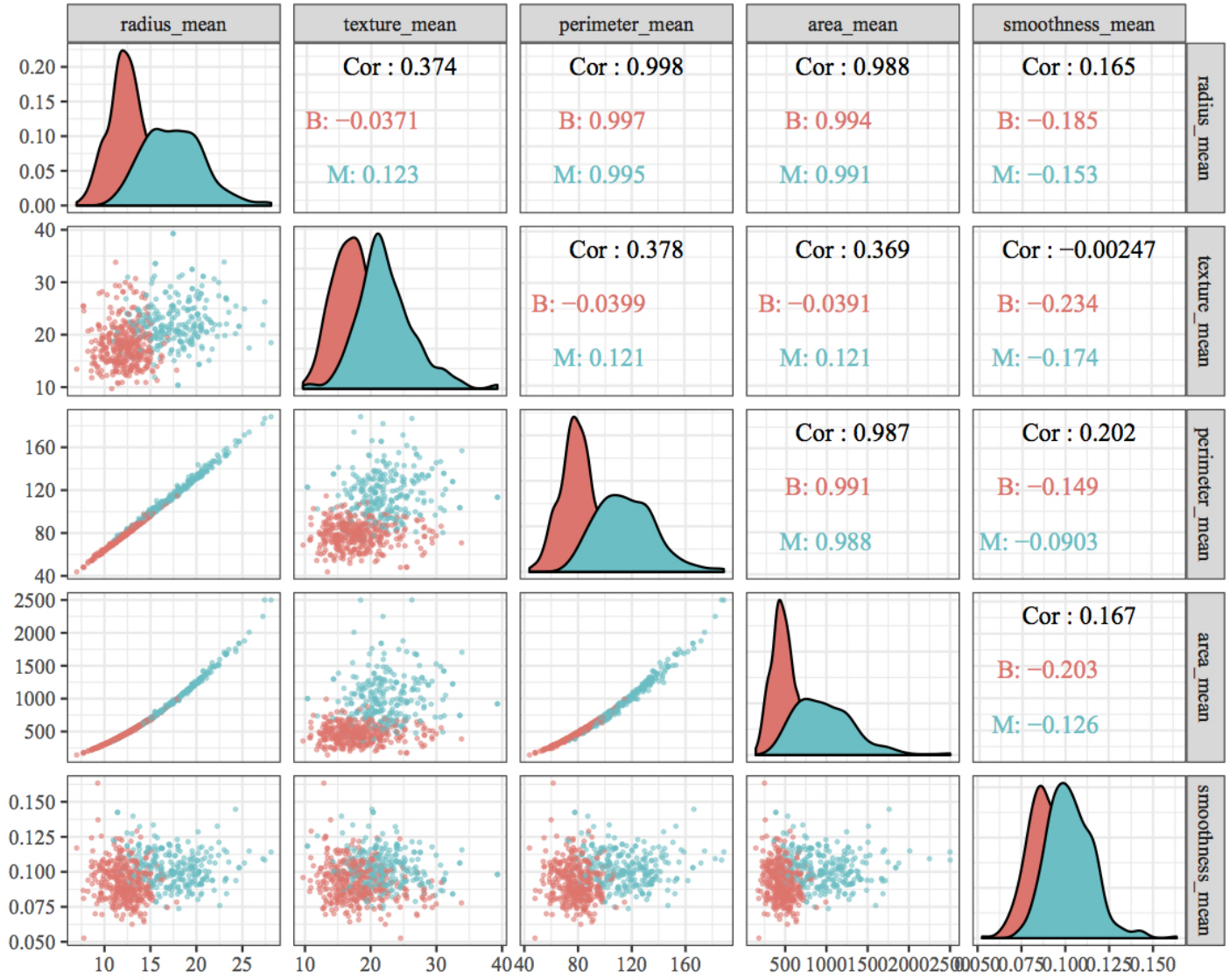
This breast cancer dataset is from Kaggle website. It originally consists of two datasets. We combined them to one dataset with 628 rows and 30 covariates by identifiers. We want to use this datasets to predict whether the breast cancer is malignant or not. These 30 covariates include 10 measures for tumors and 3 statistics for each measure. The 10 measures are radius, texture, perimeter, area, smoothness, compactness, concavity, concave.points, symmetry and fractal_dimension. The covariate 1 to 10 measure means. The covariate 11 to 20 measure standard errors. The covariate 21 to 30 measure the worsts. The table shows basic statistics for each covariate.

| mean | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
|---|---|---|---|---|---|---|---|---|---|---|
| min | 6.98 | 9.71 | 43.79 | 143.5 | 0.05 | 0.02 | 0.00 | 0.00 | 0.11 | 0.05 |
| max | 28.11 | 39.28 | 188.50 | 2501.0 | 0.16 | 0.35 | 0.43 | 0.20 | 0.30 | 0.10 |
| median | 13.65 | 19.32 | 88.08 | 572.9 | 0.10 | 0.10 | 0.070 | 0.04 | 0.18 | 0.06 |
| mean | 14.47 | 19.80 | 94.32 | 687.7 | 0.10 | 0.11 | 0.10 | 0.05 | 0.18 | 0.06 |
| se | V11 | V12 | V13 | V14 | V15 | V16 | V17 | V18 | V19 | V20 |
| min | 0.11 | 0.36 | 0.76 | 6.80 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 |
| max | 2.87 | 4.89 | 21.98 | 542.2 | 0.03 | 0.14 | 0.40 | 0.05 | 0.08 | 0.03 |
| median | 0.34 | 1.15 | 2.39 | 26.44 | 0.01 | 0.02 | 0.03 | 0.01 | 0.02 | 0.00 |
| mean | 0.43 | 1.23 | 3.04 | 44.07 | 0.01 | 0.03 | 0.03 | 0.01 | 0.02 | 0.00 |
| worst | V21 | V22 | V23 | V24 | V25 | V26 | V27 | V28 | V29 | V30 |
| min | 7.93 | 12.02 | 50.41 | 185.2 | 0.07 | 0.03 | 0.00 | 0.00 | 0.16 | 0.06 |
| max | 36.04 | 49.54 | 251.2 | 4254.0 | 0.22 | 1.06 | 1.25 | 0.29 | 0.66 | 0.21 |
| median | 15.34 | 25.79 | 101.0 | 719.4 | 0.13 | 0.22 | 0.25 | 0.11 | 0.28 | 0.08 |
| mean | 16.75 | 26.35 | 110.6 | 934.7 | 0.13 | 0.26 | 0.29 | 0.12 | 0.29 | 0.08 |
| M: 357 B: 271 | | | | | | | | | | |

We use “M” representing Malignant and “B” for Benign. The heatmap shows the correlation between 30 covariates.




This pairs plot shows the relationship for pairs of mean covariates.




## 1.2 Methods

We applied some dimension reduction and clustering techniques including principal component analysis ,multidimensional scaling , k-means, Gaussian mixture mode. Further, we conduct knock-off and LASSO for feature selection. Finally, we applied different classification methods including Support Vector Machine with different kernels, Random Forest etc. To reduce false negative error, we also applied XGBoost and conducted parameters tuning.

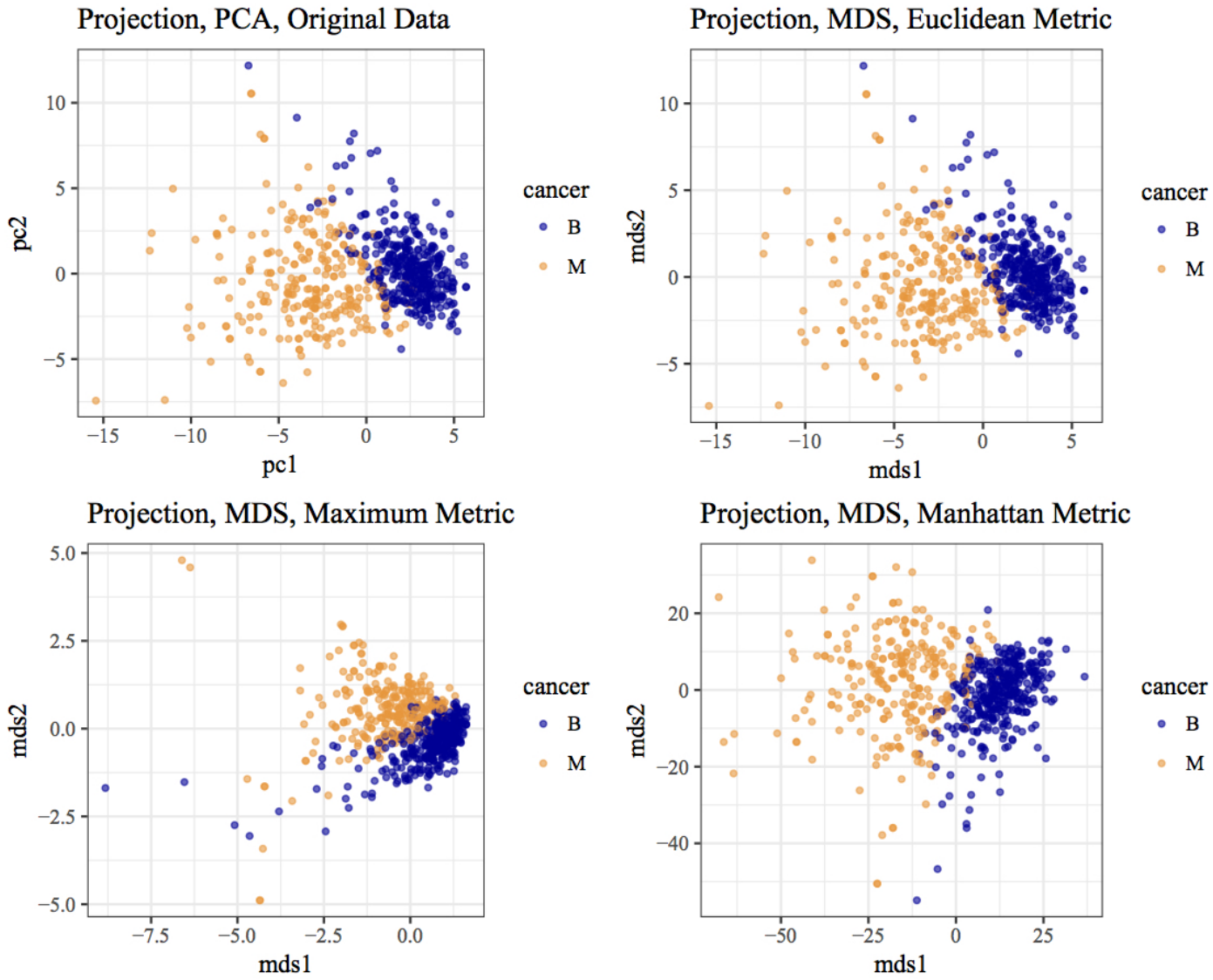
# 2 Visualizations

## 2.1 Dimension Reduction

Since the variables are in different scales, correlation was used for PCA. Two principal components are able to explain 63.5% of the variance. To achieve 90% explanation of the

variance we need seven principal components. For the sake of visualization, we only used two principal components.

To perform MDS, the data were first scaled. Similar to PCA, we only used 2-dimensional MDS for visualization. This figure represents the datasets projected on the first two PCs and MDs with different methods of distance.



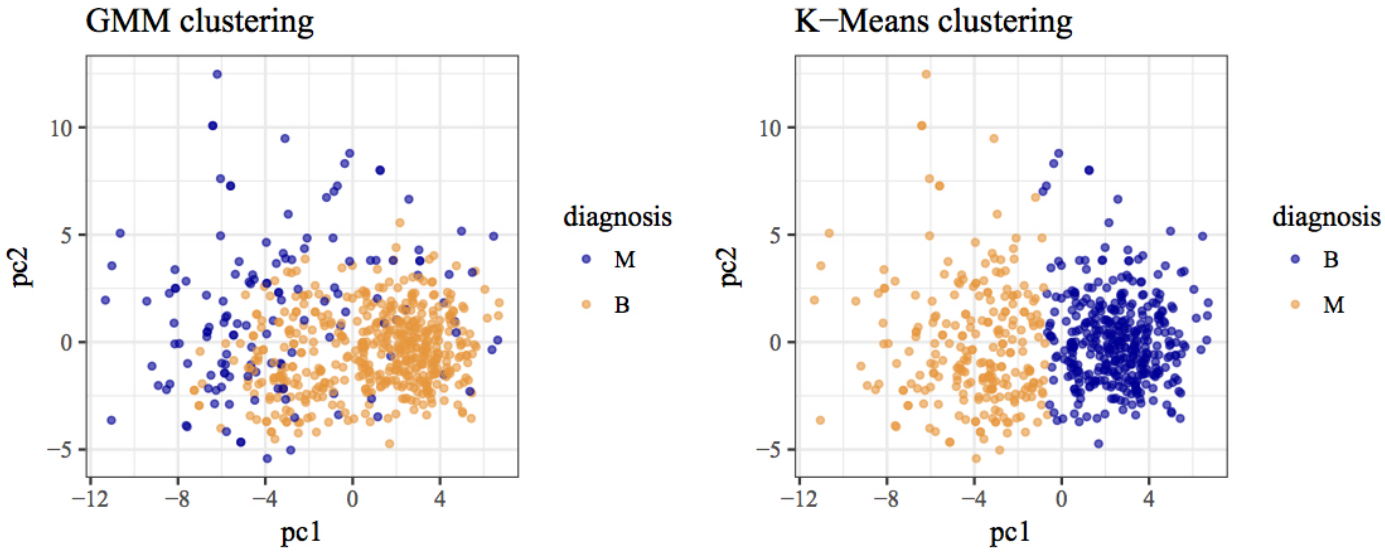

Under the Euclidean distance, the solution of MDS is the same as PCA. This table reports the STRESS value for each distance, which denoted the proportion of the distance unexplained by MDS.

| Distance | STRESS(%) |
|---|---|
| Euclidean | 28.5 |
| Manhattan | 22.8 |

| Distance | STRESS(%) |
|---|---|
| Maximum | 40.2 |

## 2.2 Clustering

From the diagram shown before, two groups of observations have a very clear clustering pattern. This reminds us to test whether the clustering results correspond to the real classification. For Gaussian Mixture Model and k-means, the clustering results are projected onto the first two components. See the figures below.




From plot, we can tell that GMM does not perform well. However, the boundary of k-means is quite desirable compared to real classification. The confusion matrix of k-means can be seen in the table. The false negative rate is 14.02% and the clustering accuracy is 91.88%.

| | Diagnosis | |
|---|---|---|
| Prediction | benign | malignment |
| benign | 344 | 38 |
| malignment | 13 | 233 |

## 2.3 Variable Selection

Basically, there are two categories of approaches to realize dimension reduction. One is variable combination, such as PCA, MDS, which have been illustrated in the last section. Another promising strategy is variable selection, such as LASSO, Knockoffs, VUSRF, etc. Variable selection has apparent advantages. First, it can facilitate us to understand and interpret the predictors. Second, it solves the dimension curse in the aspects of both model and computation. Third, sometimes, it can even achieve a better performance than the whole model. Therefore, we are encouraged to perform dimension reduction by variable selection. For Knockoffs, the idea is to generate Markov Blanket of Y and control the false discovery rate (FDR); LASSO applies L1 penalty and shrinks some coefficients to 0 to realize variables selection; VUSRF has a three-step variable selection procedure based on random forest. First step is dedicated to eliminate irrelevant variables from the dataset. Second step aims to select all variables related to the response for interpretation purpose. Third step refines the selection by eliminating redundancy in the set of variables selected by the second step, for prediction purpose.

To select the best variable selection approach, we **proposed** a randomized variable selection method. The algorithm is as follow:

procedure RANDOMIZED VARIABLE SELECTION: 1. initializematrix, withnrow=M, ncol= number of variables 2. for method i=1,...,M for j=1,...,N 3. randomly pick p percent data apply approach i and obtain coefficient C 4. save the index of non-zero coefficient in C as I matrix[i,k] = matrix[i,k] + 1 $k \in I$ 5. Choose approach m, such that maxvar(matrix[m,])

The randomized variable selection result is in the table. From the frequency of appearance, we see an elbow for each approach.

| Variable Rank | LASSO | Frequency | Knockoffs | Frequency |
|---|---|---|---|---|
| 1 | texture_worst | 1.00 | radius_worst | 0.57 |
| 2 | smoothness_worst | 0.98 | perimeter_worst | 0.56 |
| 3 | symmetry_worst | 0.97 | concave.points_worst | 0.41 |
| 4 | radius_se | 0.95 | texture_worst | 0.31 |

| Variable Rank | LASSO | Frequency | Knockoffs | Frequency |
|---|---|---|---|---|
| 5 | concave.points_worst | 0.94 | concave.points_mean | 0.24 |
| 6 | radius_worst | 0.93 | smoothness_worst | 0.21 |
| 7 | fractal_dimension_se | 0.91 | texture_mean | 0.10 |
| 8 | compactness_se | 0.89 | fractal_dimension_mean | 0.08 |
| 9 | concavity_worst | 0.86 | smoothness_se | 0.08 |
| 10 | concave.points_mean | 0.82 | radius_se | 0.07 |
| 11 | smoothness_se | 0.82 | concavity_se | 0.07 |
| 12 | concavity_mean | 0.67 | symmetry_worst | 0.06 |

Also, we used non-randomized approaches with cross-validation as reference to justify the randomized variable selection result (no order). We can observed that the most of variables are same for the random variables selection and no order cross-validation variables selection for each approach. Here, we set the false discovery rate as 0.3 for Knockoffs. Also, we tuned the penalty parameter for LASSO to shrink the coefficients. We ignored VUSRF since we will use Random Forest as classifier later. First, we can observe that the variable select from two random approaches are somehow different. Second, since the variance of Random LASSO is much larger than Knockoffs, we prefer the variables selected by Random LASSO.

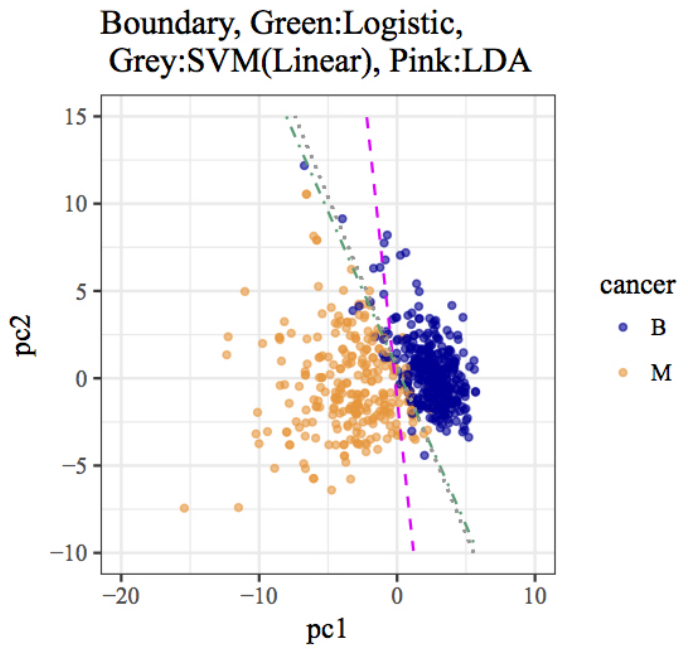
| Variable Rank | LASSO | Knockoffs |
|---|---|---|
| 1 | concave.points_mean | concave.points_mean |
| 2 | radius_se | symmetry_mean |
| 3 | smoothness_se | fractal_dimension_mean |
| 4 | symmetry_se | radius_worst |
| 5 | fractal_dimension_se | area_worst |

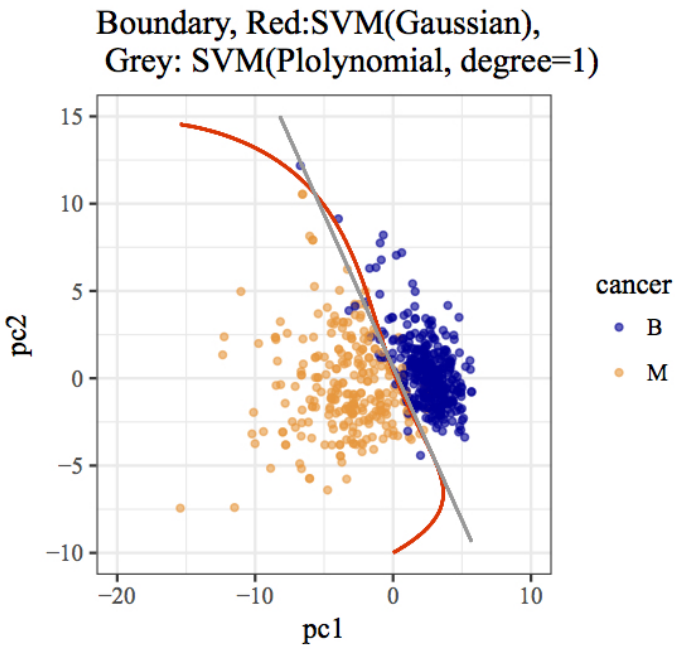
| Variable Rank | LASSO | Knockoffs |
|---|---|---|
| 6 | radius_worst | compactness_worst |
| 7 | texture_worst | concavity_worst |
| 8 | perimeter_worst | concave.points_worst |
| 9 | concave.points_worst | perimeter_se |
| 10 | symmetry_worst | smoothness_worst |
| 11 | fractal_dimension_mean | compactness_worst |
| 12 | smoothness_worst | symmetry_worst |

## 2.4 Classification

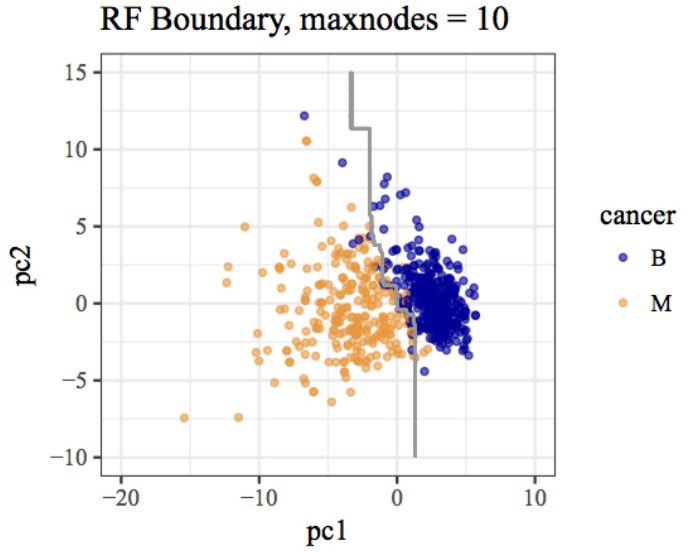
In this section, we first applied different classifiers on the first two principal components and then conduct logistic regression, SVM, random forest, XGBoost on three different version of data. ### Classification on the first two principal components for visualization In the beginning of our research, we conducted different classification methods on the first two principal components for visualization. From MDS projection figure, it seems the data is linear separable in the first two components. After tuning relative parameters on the different classifiers, we will conclude the results in the section 3.

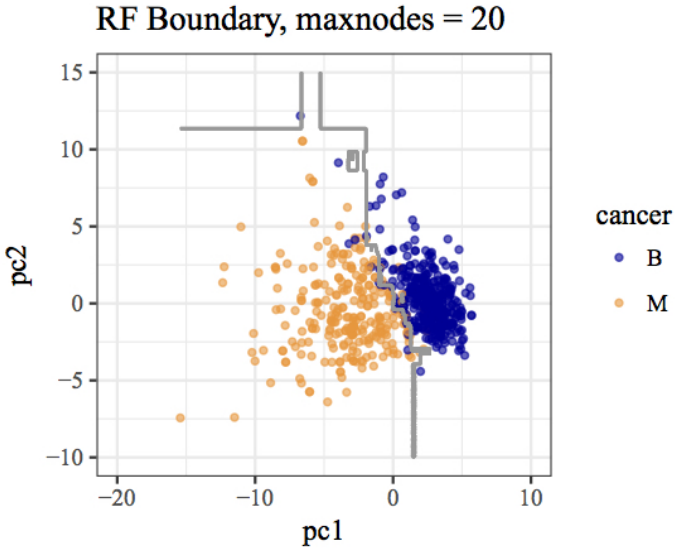
The left panel shows different linear boundary including LDA, logistic regression and soft-margin SVM. Logistic regression and SVM give the best prediction while LDA performs relatively bad, which suggests that the two classes may have different covariance or not follow multivariate normal distribution. The right panel shows the comparison between polynomial ker- nel(degree = 1) and gaussian kernel in SVM. After tuning the parameter, R reveals that degree = 1 gives the smallest CV error.

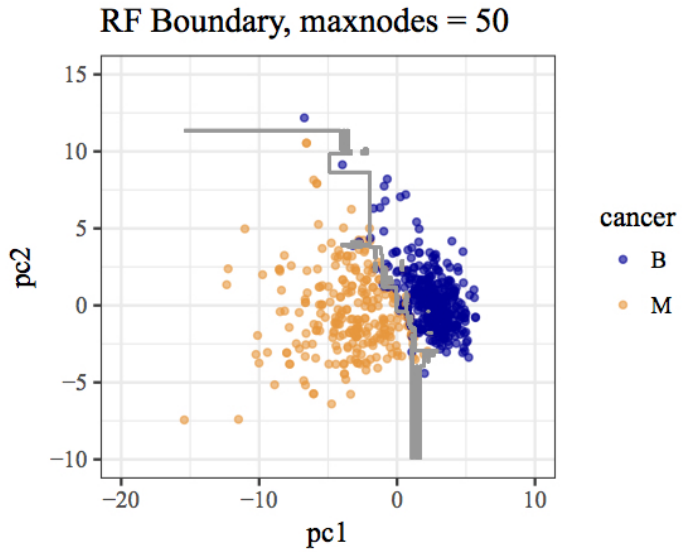


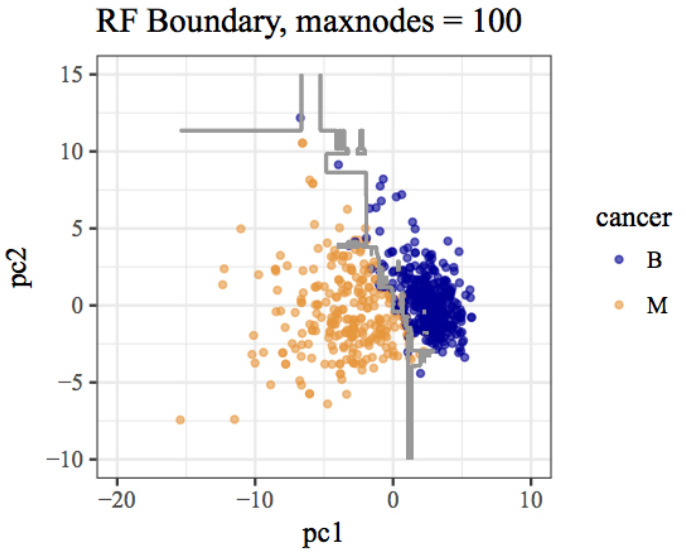




This is the boundary of random forest with different maximum number of nodes.



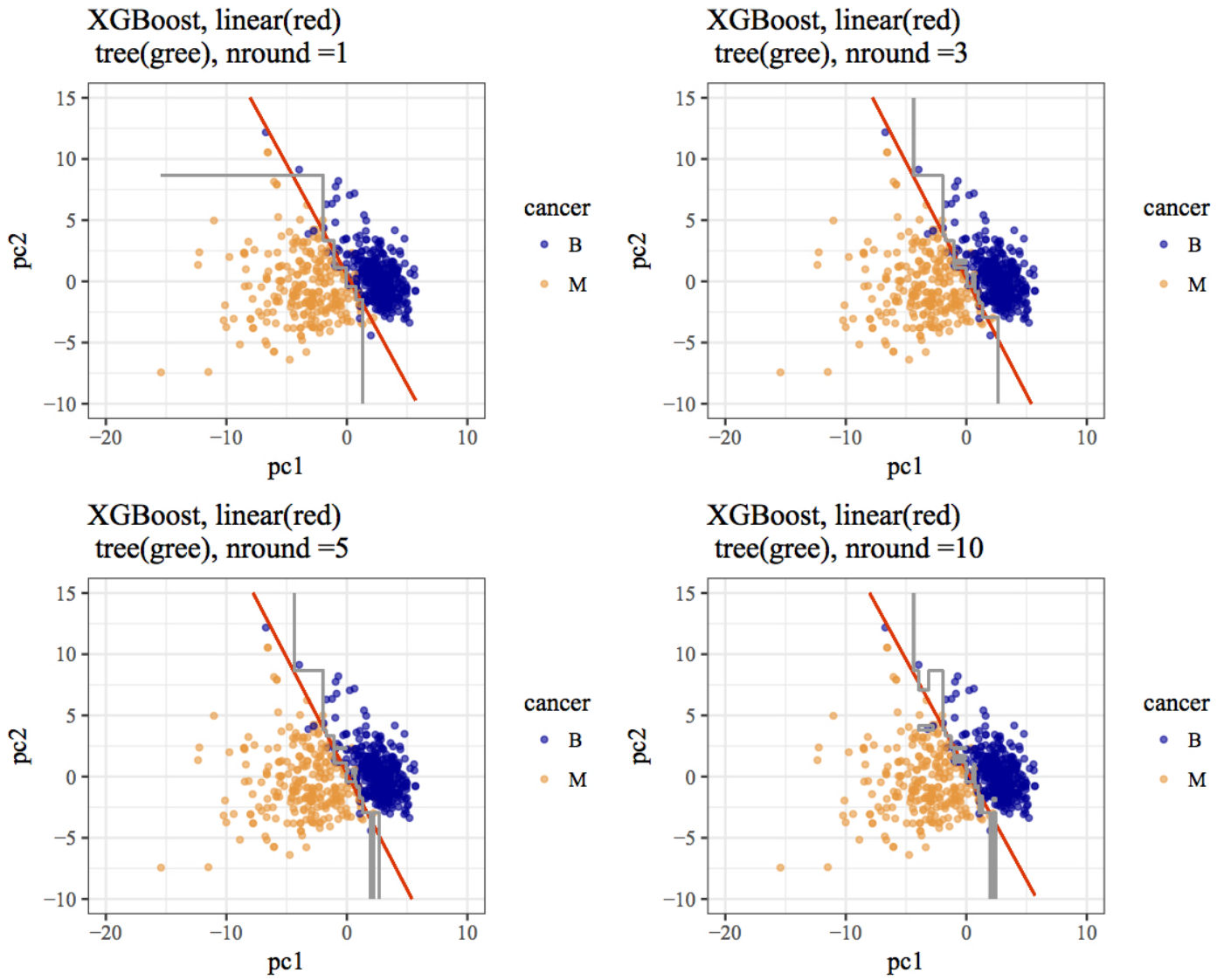
## 2.5 SVM and Random Forest on different versions of data

We mainly focused on two classifiers: SVM and Random Forest. We implemented the two classifiers on three different version of data: 1. scaled data (scale the original data) 2. log-transformed data (transform features using $log(x_{ij} + 1))$ 3. variable selection data (the 12 variables selected from section 2) Besides using accuracy as the evaluation, we also want to find the classifier which gives the best prediction on malignant breast tumor because breast cancer is a deadly disease and we want to minimize the false negative error. We use 5-fold cross- validation to tune related parameters in SVM and random forest. All the results will be attached in section 3.

## 2.6 XGBoost

there are two methods behind XGBoost, besides the tree-based one we just introduced, there is also a GLM version, which optimizes the objective function using regularization L1/L2. In this, the subsequent models are built on residuals generated by previous iterations. In this paper we implement both methods and compare them. To have an intuitive feeling, we still see how this works on 2-d data. The boundary is sketched for different nrounds (which is the number of trees we fit.), see Figure 6 From the diagrams, clearly large nrounds can lead to overfitting for tree-based model, while linear boundary is quite insensitive to this parameter. Later we will also see this by errors. After this, we tune parameters on the 30-dimension data. The parameter space of interest is: * nrounds the max number of iterations * eta control the learning rate: scale the contribution of each tree by a factor of 0 < eta < 1 when it is added to the current approximation. * gamma minimum loss reduction required to make a further partition on a leaf node of the tree * max_depth maximum depth of a tree * min_child_weight minimum sum of instance weight (hessian) needed in a child * subsample subsample ratio of the training instance * colsample_bytree subsample ratio of columns when constructing each tree

To make the tuning more reasonable, after some attempt, we decided to fix parameter nrounds to be 100, gamma to be 0 and min_child_weight to be 1. Meanwhile, we tune the other four parameters pairwise. Since max_depth and gamma together control the size of a tree, while subsample and colsample_bytree have similar meaning. This is XGBoost boundary plot on the first two PCs.




# 3 Results

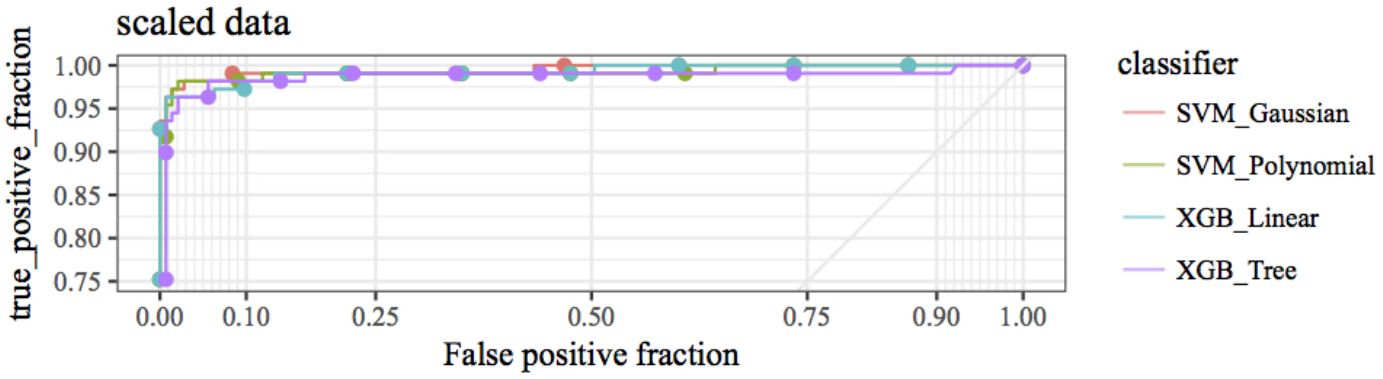
## 3.1 Sensitivity and Recall

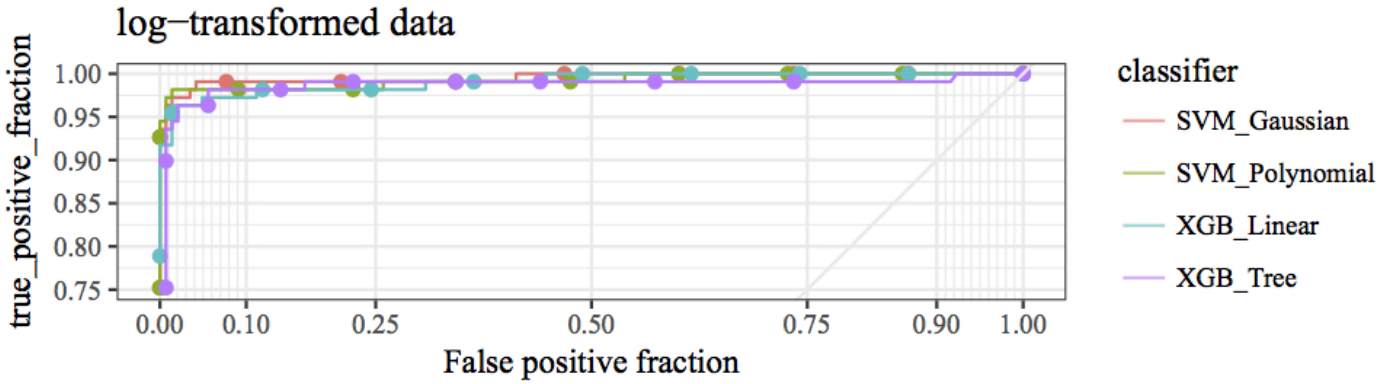
Sensitivity (or Recall / True positive rate) is critical in this scenario. The sensitivity and accuracy for three types of data are in the table.

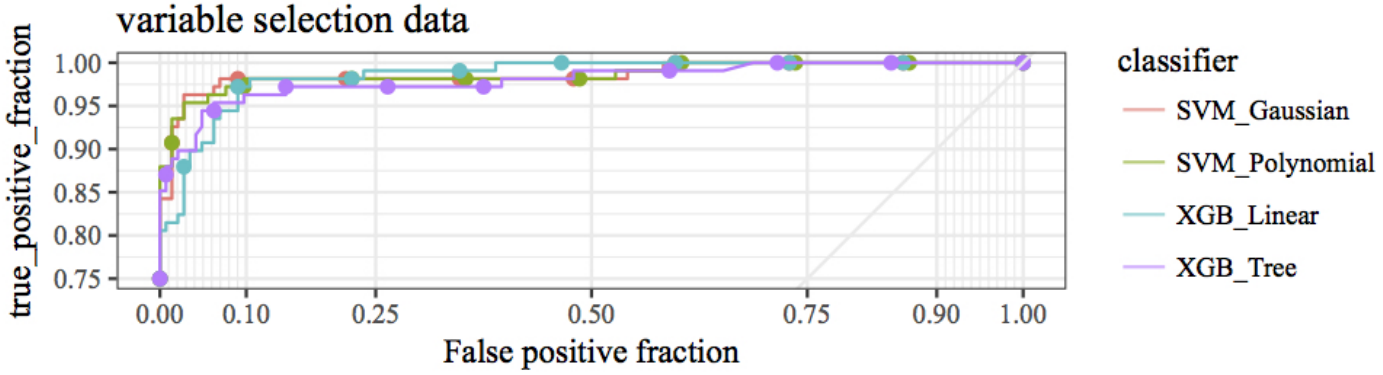
| | scale data | | log transformation data | | variable selection data | |
|---|---|---|---|---|---|---|
| | Accuracy | Recall | Accuracy | Recall | Accuracy | Recall |
| SVM-Gaussian | 0.9524 | 0.8981 | 0.9603 | 0.9167 | 0.9524 | 0.8519 |
| SVM-Polynomial | 0.9762 | 0.9537 | 0.9722 | 0.9630 | 0.9643 | 0.9352 |
| Random Forest | 0.9484 | 0.9286 | 0.9484 | 0.9196 | 0.9563 | 0.9375 |
| XGBoost-Tree | 0.9762 | 0.9921 | 0.9683 | 0.9921 | 0.9643 | 0.9683 |
| XGBoost-Linear | 0.9683 | 0.9762 | 0.9841 | 0.9841 | 0.9365 | 0.9127 |

## 3.2 ROC curve of four classifiers on three data version

The set of graphs is SVM-Gaussian, the four classifiers' - SVM-Polynomial, XGBoost-Tree and XGBoost-Linear, ROC curves on three different versions of data.



The table shows the area under the ROC curve(AUC), the closer to one, the better the classifier is. The value of AUC indicates that SVM performs the best but the difference is not in a big scale. However, in terms of recall rate, XGBoost outperforms any other classifier.

| | SVM-Gaussian | SVM-Polynomial | XGBoost-Tree | XGBoost-Linear |
|---|---|---|---|---|
| scaled data | 0.9922 | 0.9947 | 0.9924 | 0.9865 |
| log-transformed data | 0.9952 | 0.9924 | 0.9910 | 0.9865 |
| variable selection data | 0.9862 | 0.9862 | 0.9855 | 0.9803 |

# 4 Conclusion

Linear XGBoost achieves the best accuracy 0.9841 on log transformation data; Tree XGBoost achieves the best recall 0.9921 on both scaled data and log transformation data. SVM-polynomial have a close performance. SVM-Gaussian have an acceptable accuracy while the recall rate is far below other classifiers. Random Forest is the worst in accuracy and performs bad in terms of recall. Also, considering about the data type, scaled data and the log-transformed data are both preferable in terms of accuracy. However, the log-transformed data outperforms scaled data in recall rate. For all the classifiers except Linear XGBoost, the variable selection data achieves the results close to log-transformed data both in accuracy and recall. Thus, it is worthy to conduct a variable selection for the computational efficiency and for the prediction of Malignant breast tumor with less medical test. In the theme of medicine, we focus on more about recall than accuracy. Hence, the best classifier is Linear XGBoost since it significantly outperforms. Besides, the best data type for the specific problem is log transformation. The reason why XGBoost has an apparent advantage over the others is that it produces a strong prediction model in the form of an ensemble of weak prediction models, plus multiple anti-over-fitting techniques. What's more, this dataset typically requires modest amount of cleaning/pre-processing, has a modest data set size and the nature of the problem is a boolean classification. Given these characteristics, gradient boosting happened to be among the most powerful classifiers to throw at such problems.