CS7641 Homework 1 – Youngsik Cho

Table of Contents

0. Introduction

This report conveys machine learning analysis with various unsupervised/classification algorithms in R software. R is a free software environment generally used for computing and statistical analysis. Libraries used and more information about the software are included in the README file.

This will introduce two different datasets for comparison in different algorithms' accuracy of how well to predict their response variables. Then, it ends with rankings of those algorithms in terms of the accuracy with different sizes of training and testing sets. In most cases, 10-folds cross-validation technique is used.

1. Data Description

As presented in the R script, two different datasets are used to represent the following algorithms: decision tree, neural network, gradient boosting model, support vector machine, and k-nearest neighbors. When choosing interesting problems/datasets following conditions are concerned: it has enough data observations to subset for training and testing, it does not have too many missing data, and it is contextually meaningful and interesting.

1.1 Dataset 1: Titanic Survival

The first dataset is called as the Titanic Survival. The dataset contains data points of each person who was on the Titanic cruise ship and about whether they survived or not. There was no statement that these data points are factual, however, the dataset itself is very interesting as I have been curious about the accident and wanting to figure out the most important features or states survivors had at the time of the accident that impacted their survival rates. The rankings of the features may help answer some questions: Did younger people survive more? Did people who paid more for their trip survive better? and so on. Also, I really enjoyed the movie. This dataset has 1309 observations with 15 features. Among those features, the 'survived' variable plays a role as response variable and the rest of quantitative or binary/integer variables serve as predictors. Irrelevant features are excluded. Therefore, the rest of attribute that help predict are: 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare', and 'embarked'. This dataset is not trivial at all since it contains enough number of data points as well as number of predictors to be used for various machine learning algorithms.

The more detailed information of the features is shown below.

1.1.1 Data Dictionary

Variable	Definition	Key
Survival	Survival	0 = No, 1 = Yes
Pclass	Ticket Class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	sex	Male/Female
Age	Age in years	

Sibsp	Number of siblings/spouses aboard the Titanic	
Parch	Number of parents/children aboard the Titanic	
Ticket	Ticket number	
Fare	Passenger fare	
Cabin	Cabin number	
Embarked	Port of embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

1.1.2 Variable Notes

- Pclass: A proxy for socio-economic status (SES)
  - 1st = Upper
  - 2nd = Middle
  - 3rd = Lower
- Age: Age is fractional if less than 1. If the age is estimated, it is in the form of xx.5.
- sibsp: The dataset defines family relations in this way.
- Sibling includes brother, sister, stepbrother, stepsister.
- Spouse includes husband, wife (mistresses and fiancés were ignored).
- Parch: The dataset defines family relations in this way
- Parent counts mother, father
- Child counts daughter, son, stepdaughter, stepson
- Some children travelled only with a nanny, therefore parch = 0 for them.
- More detailed information can be found at https://www.kaggle.com/c/titanic.

1.2 Dataset 2: Breast Cancer

The second dataset is regarding whether a patient has a breast cancer or not. This breast cancer dataset was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. For classification purpose, the dataset is divided into 10 attributes and a response variable. This dataset originally has 699 data points from 8 different instances in which dates and time vary, including 16 observations with missing data. All the rows that contains missing data and NA values are omitted before used for classification algorithms, and therefore the number of data points remains to 683. It is far greater than the number of attributes, and that makes this dataset quite suitable for any classification problem. In terms of context, the attributes are in regard of breast cancer cells, and they include the size of the cell, the thickness of clump, uniformity of cell size, and so on. The goal here is to correctly classify if a patient has breast cancer cells or not by only using characteristics of the cells. Here is more information about the attributes.

```
#  Attribute                     Domain
   -- -----------------------------------
   1. Sample code number            id number
   2. Clump Thickness               1 - 10
```

```
  3. Uniformity of Cell Size       1 - 10
  4. Uniformity of Cell Shape      1 - 10
  5. Marginal Adhesion             1 - 10
  6. Single Epithelial Cell Size   1 - 10
  7. Bare Nuclei                   1 - 10
  8. Bland Chromatin               1 - 10
  9. Normal Nucleoli               1 - 10
 10. Mitoses                       1 - 10
 11. Class:            2 for benign, 4 for malignant
```

Bare nuclei corresponds to naked nuclei, which is typically seen in cell degeneration. More information can be found at: https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.names.

Both datasets are divided into train and test sets. The code is set in such a way that the fraction of the train and test sets vary, and the corresponding results get recorded. Using a for loop, the datasets are tested based on 20%, 40%, 60%, and 80% as training sizes.

2. Algorithms

As mentioned above, five different classification algorithms are used to be compared: decision tree, neural networks, boosting, support vector machine (SVM), and k-nearest neighbors (KNN). In this section, all those five are defined and how those are used in the datasets is shown.

2.1 Decision Tree

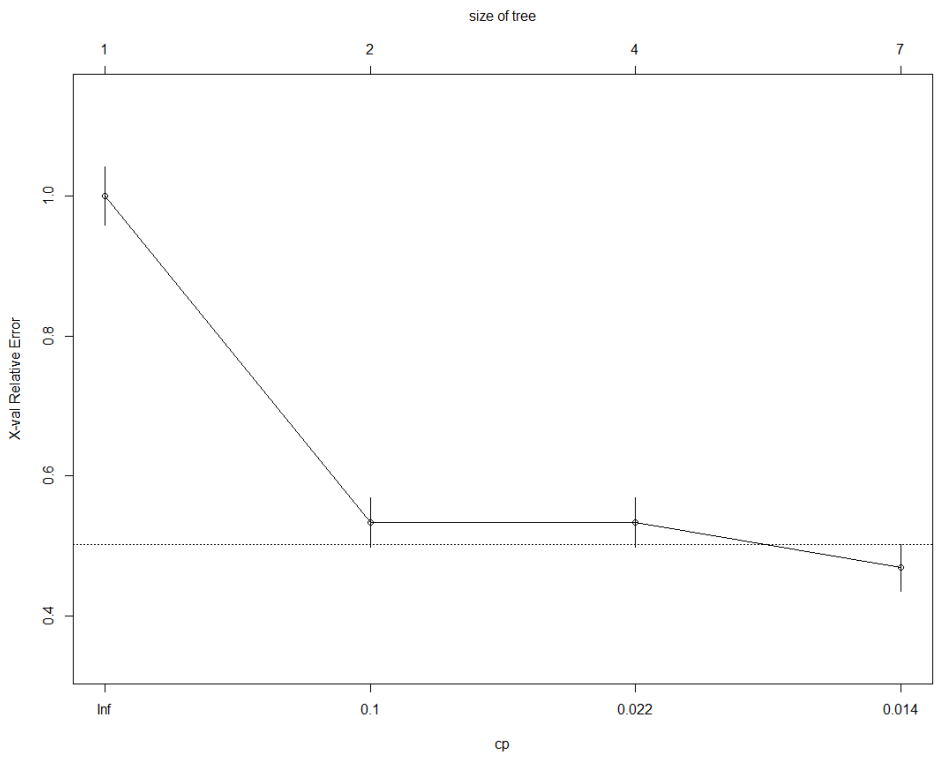
The decision tree algorithm is a decision support tool that a tree-like depiction or decision models helps data points find own way to certain consequences or nodes. It consists of many decision points where data are split depending on their conditions and the requirements of those points. As a general result, all the data points are subset to final nodes. The decision points are decided based on their significance to the data. The significance can be measured by how much the decision can impact the response variable.

The decision tree classification algorithm is implemented by using 'rpart' package. The basic steps are the same for both datasets. After fitting the tree to the training set, variable importance factors are analyzed. It demonstrates which factors or attributes are the most contributing factors to the response.

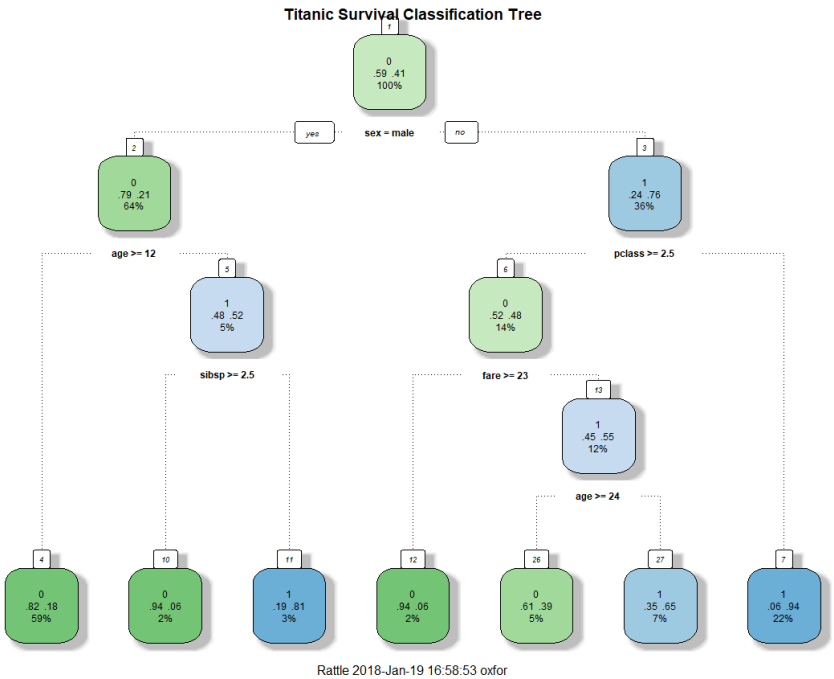
For the titanic dataset, the most important factors are: 'sex', 'fare', 'pclass', and 'sibsp'.

```
       sex       fare     pclass      sibsp        age      parch   embarked
118.665003  38.684233  30.661284  16.380973  16.207581  11.211472   8.894709
```

Then, the complexity parameters are analyzed. The graph below represents how relative error and complexity parameter decreases as the size of tree or the number of splits increases. This trend is very general.



The smallest relative error turns out to be 0.42183 with complexity parameter of 0.01. By using the 'rattle' package, the tree can be visualized in more intuitive way:

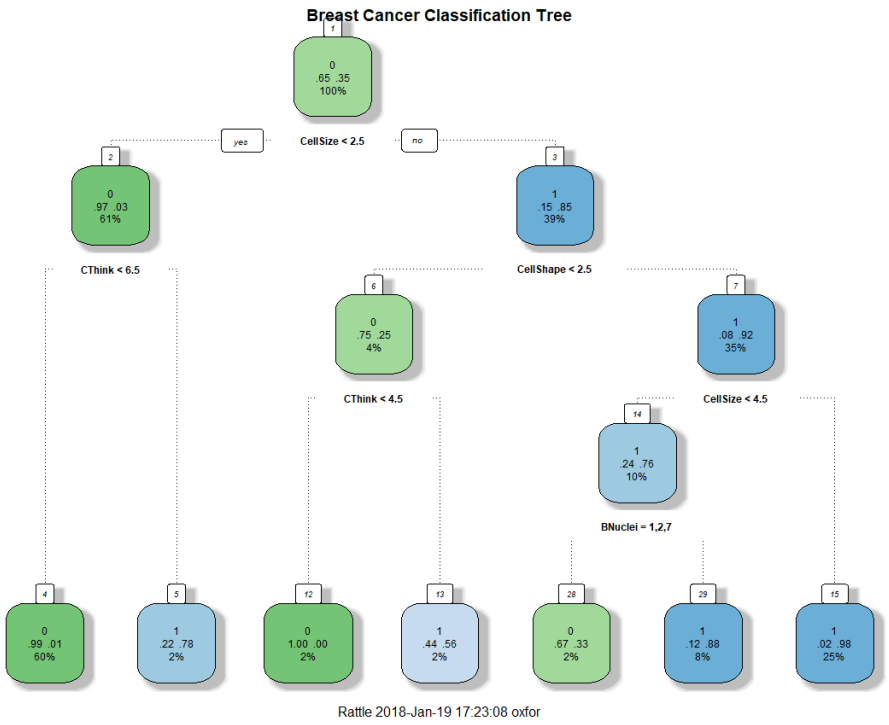


Each node box displays the classification, the probability of each class at that node (i.e. the probability of the class conditioned on the node) and the percentage of observations used at

that node. This tree results in 7 nodes total. The green nodes represent 0 (not survived), and the blue nodes represent 1 (survived). However, it still does not mean the all data points in the nodes are fixed 0 or 1. It means most them are 0 or 1. That is why the intensity of the colors differ, to show that the more intense color means the higher percentages of the final responses of the data points in the nodes. As the variable importance factors suggest, those most important factors are used for branching decisions. It seems people younger than 24 had better chance of surviving, and so on. However, this tree might be having too many leaves. Therefore, pruning is performed on the tree, but the resulting tree does not change.

Then, the tree is used to predict on both train and test datasets.

For the breast cancer dataset, the same steps have been placed to fit the tree. The only difference in this tree compared to the one above is that the complexity parameter is limited to 0.005. Therefore, the number of splitting is limited to 6 times. Because of the restrictions, the branching happens when necessary.



Also in this case, the tree does not change by pruning since all the important variables have been used for decisions.

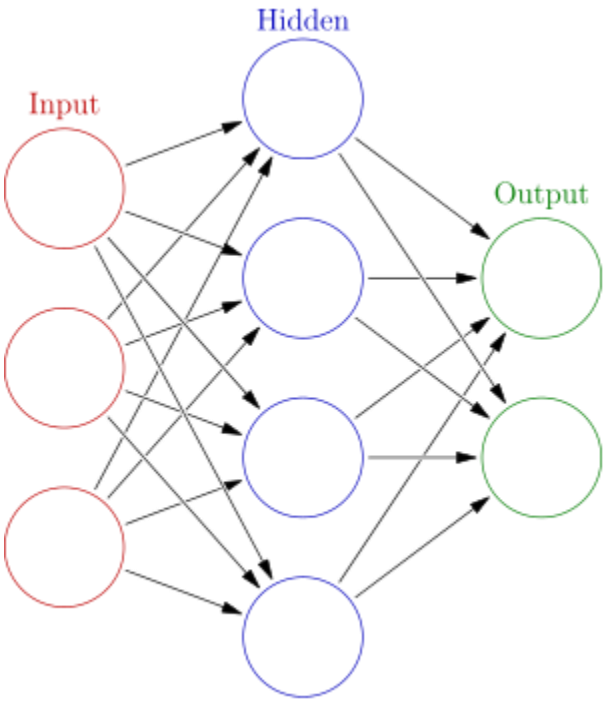
```
  CellSize  CellShape SECellSize    BNuclei NormNuclei     BlChrom     CThink     MargAdh     Mitoses
178.367796 150.626423 134.565182 134.009021 122.000604 121.941685  13.318355   1.412356   1.018519
```

Then, the fitted model is tested on both training and testing sets about how well the predictors in both sets predict the survival. The final results are discussed in the later section.

2.2 Neural Network

Neural networks or artificial neural networks or connectionist systems are computing algorithm inspired by biological animal brains, where they learn or progressively improve in predicting tasks by considering various examples repetitively. They are interconnected group of nodes very similar to neurons in brains. Data points undergo input, hidden, and output nodes in order.

For neural network algorithm, 'caret' package is used. After factorization of the data, 10-folds cross validation is used with threshold of 0.95 for both datasets by trainControl method. Then the neural network model is trained with 2 units of hidden layer as suggested in this post: https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw.



For both neural network models for two datasets, plotting is not very intuitive and therefore skipped. The plot usually depicts neurons and layers like the diagram above. In short, every attribute takes place in the input and hidden nodes, and the response appears in the output nodes. The arrows represent the connections between those nodes and learning processes. More details are explained in http://neuralnetworksanddeeplearning.com/.

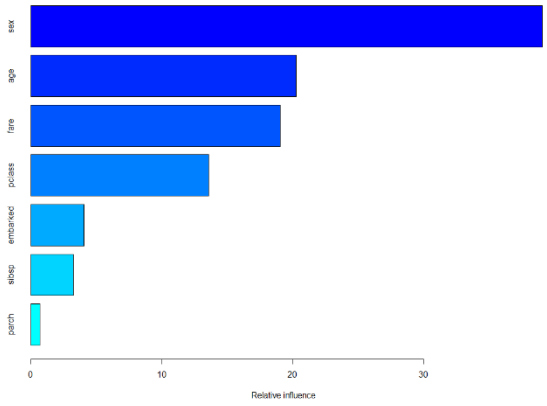
The results are discussed in the later section.

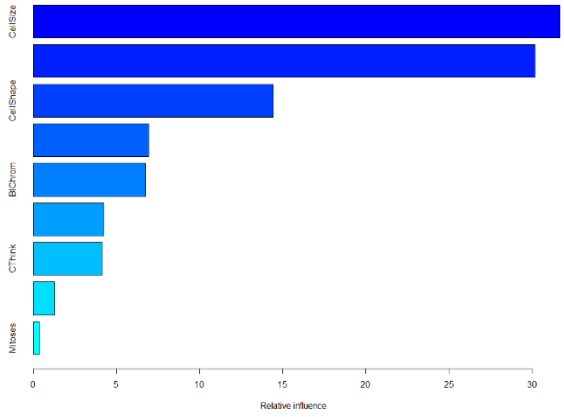
2.3 Boosting

Boosting means more refined version of weak prediction models like the decision tree algorithms. There are multiple boosting methods, but gradient boosting is used in this section. The idea of gradient boosting is interpreted to optimize the cost function within a decision tree. It can also be perceived as an iterative functional gradient descent algorithm, which optimizes a

cost function over a functional space by iteratively choosing a weak hypothesis that points in the negative descent. More detailed information can be found here: https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d.

For gradient boosting algorithm of decision trees, 'gbm' package is used. The 'gbm' method is used to fit the tree based on 10-folds cross validation and 'adaboost' distribution, which is best to use when having binary response variable. Therefore, it is used for both datasets. Also, 10000 trees are used with 0.0005 shrinkage and 5 interaction depths. The number of trees is equivalent to the number of iterations. It is generally set to have enough to find the optimal number of trees. The shrinkage represents the learning rate and the interaction depth is used as a form of pruning. The interaction depth is set to 4 for both datasets as suggested in the 'rpart' method. Plotting also here is very not intuitive to draw thousands of trees and therefore skipped. Nevertheless, relative influence of attributes for both datasets is measured.

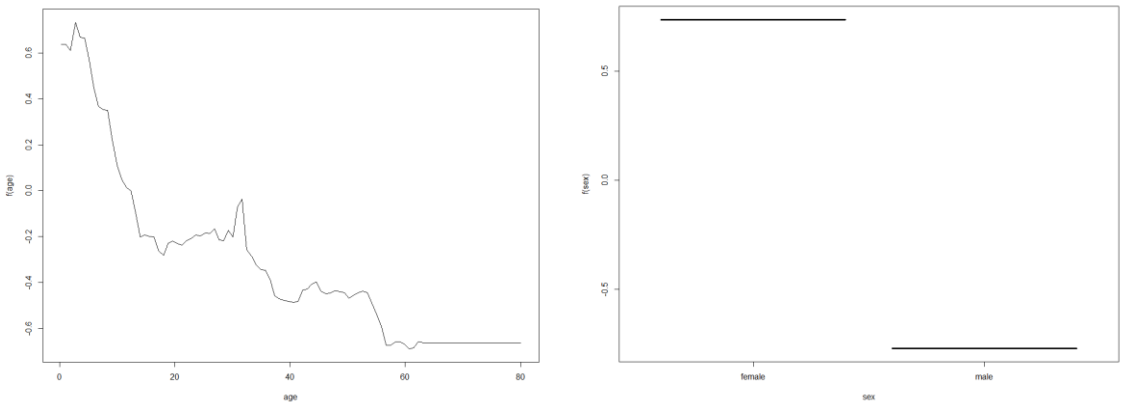




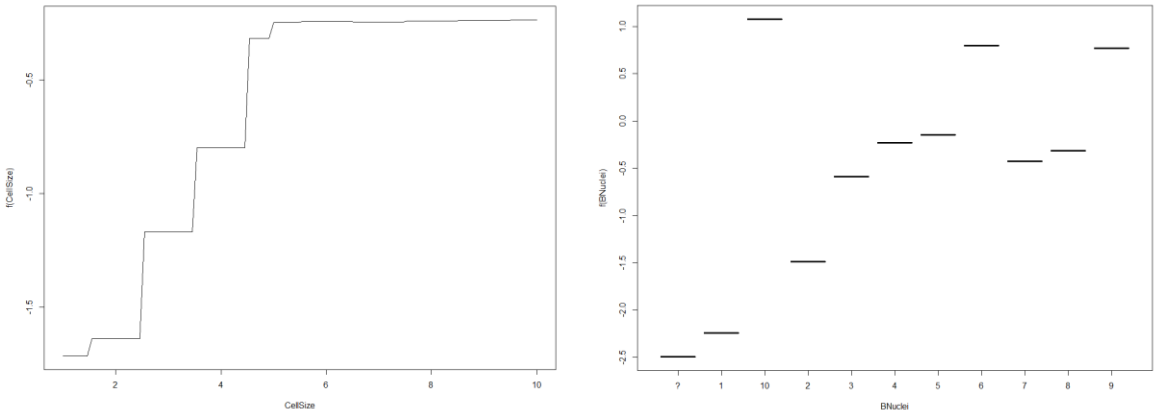
var	rel.inf
sex	39.0573247
age	20.2743796
fare	19.0611235
pclass	13.5949278
embarked	4.0599655
sibsp	3.2667909
parch	0.6854881

var	rel.inf
CellSize	31.6781448
BNuclei	30.1729781
CellShape	14.4378845
NormNuclei	6.9238434
BlChrom	6.7363372
SECellSize	4.2399956
CThink	4.1469683
MargAdh	1.2833785
Mitoses	0.3804697

Just like in the single decision tree algorithms, these plosts suggest the most influencial features to the response. The most significant and impactful attributes to the survival in the Titanic dataset is sex, age, fare, ticket class of the person, and those for the Breast Cancer dataset are size of the cell, number of bare nuclei, and the shape of the cell.



For the Titanic dataset, these graphs can signify that females had better chance to survive than males and that the survival rate and age are inversely related. The younger aged people seems to have higher survival rates.



For the breast cancer dataset, these graphs state that the size of the cell and the number of bare nuclei will increase the chance of getting diagnosed with breast cancer.

Then, by using the optimal number of trees, the gradient boosted decision trees are used to predict response variables for both train and test sets. Again, the accuracies of those models are discussed later.

2.4 Support Vector Machine (SVM)

Support vector machines (SVMs) are also one of the supervised machine learning algorithms that are used to classify or predict a response based on multiple predictors. Those can be also used for regression analysis. This page demonstrates detailed examples with principles behind the concept: http://scikit-learn.org/stable/modules/svm.html.

Also for SVM models, parameters are tuned first by 10-folds cross validation. Two parameters are mainly tuned: gamma and cost. The hyperparameter gamma controls the tradeoff between error due to bias and variance in your model, and the cost controls the cost of misclassfication

on the training data. These parameters define the balance between 'not too strict' and 'not too loose'. The goal of the tuning is to find the optimal values for both parameters. For both problems, the tuning step is commented due to the runtime. Also, 'radial' (default) kernel and 'polynomial' kernel are used.

```
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  10
      gamma:  0.01
    epsilon:  0.1


Number of Support Vectors:  408
```

```
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  polynomial
       cost:  10
     degree:  3
      gamma:  0.01
     coef.0:  0
    epsilon:  0.1

Number of Support Vectors:  680
```

```
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  12
      gamma:  0.01
    epsilon:  0.1


Number of Support Vectors:  221
```

```
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  polynomial
       cost:  12
     degree:  3
      gamma:  0.01
     coef.0:  0
    epsilon:  0.1

Number of Support Vectors:  340
```

The two images on the top represent two different SVM models for the Titanic dataset, and the others are for the Breast Cancer dataset. Other parameters beside cost and gamma are not tuned.

Plotting as well as interpreting the plots are very difficult in both problems since both have too many dimensions and support vectors. Therefore, it is also skipped here. For a two-dimensional problem, it is much easier to see how existing data points are classified and new points are predicted. The basic idea is to divide the data points by a line if in a two-dimensional space, by a plane in a three-dimensional space, etc.

After tuning, those fitted SVM models are used to predict reponse variable for both train and test data.

2.5 K-Nearest Neighbor (KNN)

KNNs are also used for both classification and regression. In this case, classification is practiced. Differently from SVM models, KNNs classify data points by their 'nearest neighbors'. A new data point is classified as a response solely depending on majority number of nearest neighbors. For more information: https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering/.

To address those problems, however, weighted KNN models are used to weigh more on neighbors that are 'nearer' than 'further' to new points. The 'kknn' package contains dedicated weighted KNN algorithms, so they are used for these problems.

Parameter tuning is also vital in KNN algorithms. By using 10-folds cross validation, a 'for loop' is used to identify the best k-value: the number of neighbors. Total of 20 k-values (initially 150) are tested for both problems and whichever k's with highest accuracy on training sets are chosen to be used for testing sets. Corresponding accuracies are recorded and discussed in the next section.

3. Final Results

	0.2	0.4	0.6	0.8
tree1.tr	0.829384	0.833729	0.837061	0.827918
tree1.te	0.784173	0.756410	0.801909	0.803738
tree2.tr	0.955882	0.930403	0.943765	0.950549
tree2.te	0.923218	0.921951	0.948905	0.963504
nn1.tr	0.829384	0.828979	0.808307	0.805054
nn1.te	0.785372	0.743590	0.792363	0.803738
nn2.tr	1.000000	0.985348	0.980440	0.979853
nn2.te	0.952468	0.956098	0.970803	0.978102
boost1.tr	0.800948	0.840855	0.829073	0.832732
boost1.te	0.769784	0.785256	0.797136	0.817757
boost2.tr	0.992647	0.978022	0.975550	0.976190
boost2.te	0.976234	0.970732	0.974453	0.985401
svm1.rad.tr	0.791469	0.805226	0.798722	0.791817
svm1.rad.te	0.781775	0.775641	0.773270	0.785047
svm1.pol.tr	0.658768	0.627078	0.615016	0.602888
svm1.pol.te	0.582734	0.575321	0.582339	0.593458
svm2.rad.tr	0.985294	0.970696	0.970660	0.968864
svm2.rad.te	0.968921	0.968293	0.974453	0.970803
svm2.pol.tr	0.742647	0.732601	0.760391	0.765568
svm2.pol.te	0.707495	0.746341	0.737226	0.802920
knn1.tr	0.791469	0.821853	0.806709	0.811071
knn1.te	0.764988	0.769231	0.806683	0.817757
knn2.tr	0.977941	0.967033	0.973105	0.968864
knn2.te	0.599634	0.597561	0.594891	0.627737

The table above is formed by the percentages of training set as columns, different algorithms on training and test sets as rows, and corresponding accuracies as cells. The trailing numbers in row names represent datasets: '1' for the Titanic and '2' for the Breast Cancer. For example, the first cell of 0.829384 indicates that with 20% training set and 80% test set from the Titanic dataset, the decision tree algorithm results in 82.9384% of accuracy on the training set. The

general trend in the table is quite obvious. As the percentage of the training set increases, so does the accuracy, and that is because the algorithm can learn more thoroughly with more examples. Also, the algorithms performed better with training set due to the concept of overfitting. In statistics, overfitting refers to 'the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably.' In other words, it is too closey fit to the training sets to deal with new random patterns in the testing sets. The difference in accuracy between training and testing sets tends to decrease as the percentage of the training set gets bigger since it becomes better at predicting with more examples as mentioned above. On the training set, the neural network 2 predicted most accurately, and suppor vector machine 1 polynomial did most poorly. On the other hand, boosting 2 performed best, and k-nearest neighbor 2 did worst.

There are many ways to further improve the performance. For example, more data points, usage of multiple algorithms together, more stringent parameter tuning, ensembling methods like boosting and baggin on traditional methods, and so on may improve the performance.

Among those algorithms, neural networks take much more time than others mostly because of number of features in datasets. Overall, the time to run all five is around 3 minutes.