Predictive Data Analytics

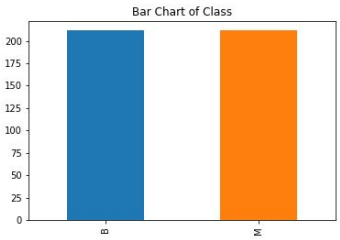
Name: Li Sun

I. TASK 1

There are 424 instances and 30 descriptive attributes in the cancerData dataset. The data type of target attribute 'Class' is 'object', and the data type of all 30 descriptive attributes is 'float64'.

In the attribute of 'Class', 'B' is mapped into '0', and 'M' is mapped into '1'. The bar charts in Figure 1.1 shows these two classes are balanced. There are 212 instances in each class.

Figure1.1 Bar Chart of Class



The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values [1]. Therefore, from the machine learning perspective, there is no significant difference between normalizing data to [0,1] vs. [0,3] range.

II. TASK 2

A. *Split Data*

By performing stratified holdout sampling and random_state = 5, the original data is split into training dataset and testing dataset, and the detailed info is shown Table 2.

Table 2.1 Split Data

Dataset	Proportion	Number of Instances
Training	2/3	282
Testing	1/3	142

B. *Decision Tree*

Decision Tree is applied to predict testing data using Entropy and Gini with level 1 to 10. Their scores are displayed in table 2.2.1 and table 2.2.2. And their corresponding accuracy plots are shown in figure 2.1. The best score for decision trees with entropy is 94.3%, and with Gini is 90.8%.

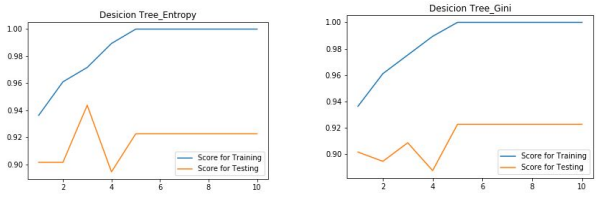
Table 2.2.1 Decision Tree with Entropy

Level	Training Score	Testing Score
1	0.936170	0.901408
2	0.960993	0.901408
3	0.975177	0.943662
4	0.989362	0.894366
5-10	1.000000	0.922535

Table 2.2.2 Decision Tree with Gini

Level	Training Score	Testing Score
1	0.936170	0.901408
2	0.960993	0.894366
3	0.975177	0.908451
4	0.989362	0.887324
5-10	1.000000	0.922535

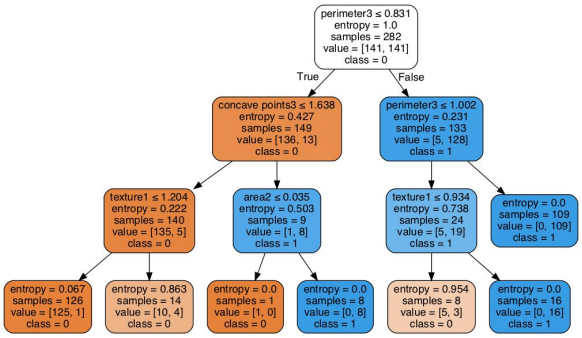
Figure 2.1 Plots of Decision Tree with Entropy and Gini



The best tree is the Decision Tree with Entropy having 3-node depth. Since our classes are balanced, the classification accuracy works well and we can use accuracy scores to find the best model. From the above charts and plots, we know the best testing score is from Entropy with level 3, which is 94.4%, with low risk of overfitting or underfitting.

The winning tree is plotted in Figure 2.2 as follows.

Figure 2.2 Visualization of Decision Tree with Entropy



III. TASK 3

KNN Classifier is applied to fit training dataset and to predict testing dataset with Euclidean or Manhattan Distance measure, with uniform or distance weight. The number of neighbors are chosen from 1 to 20.

A. *The Analysis on Accuracy Score*

In the method of KNN with Euclidean and Uniform Weight, the best testing scores are 97.9%, with k= 3 or 5.

In the method of KNN with Manhattan and Uniform Weight, the best testing scores are 98.6%, with k = 1.

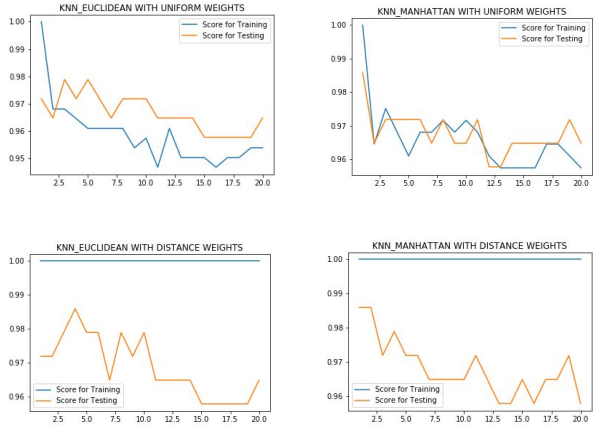
In the method of KNN with Euclidean and Distance Weight, the best testing scores are 98.6%, with k = 4.

In the method of KNN with Manhattan and Distance Weight, the best testing scores are 98.6%, with k = 1 or 2.

B. *The Accuracy Score Plots of each KNN Classifier*

Figure 3.1 displays the accuracy scores of each KNN classifier.

Figure 3.1 Score Chart of KNNs



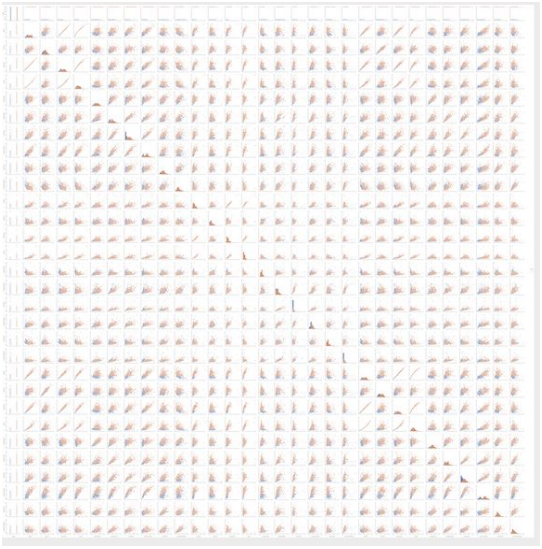
C. *Conclusion*

There are several different KNN classifiers which achieve the highest score of 98.6%. But all of them are with a small value of k, which means that noise will have higher influence on the result. So a suitable k with a reasonably high score should be a best solution.

In order to decide which investigated distance measure is better, a SPLOM is visualized as Figure 3.2. The two groups of classes do not seem to have circular shapes, which would help our Euclidean-based KNN shine. Also distance weighted method is more robust to noise than uniformed weight. There are also some suggestions about choice of k, for instance k should not be too small or too large, k value should be odd, and A simple approach to select K is set k=n^(½) [2], in this case, 282^(½) is around 16.

Overall, from the above discussion, I recommend that the winning KNN classifier is Euclidean with distance weight and k = 9, which has an accuracy score = 97.2%.

Figure 3.2 SPLOM of Normalized Dataset

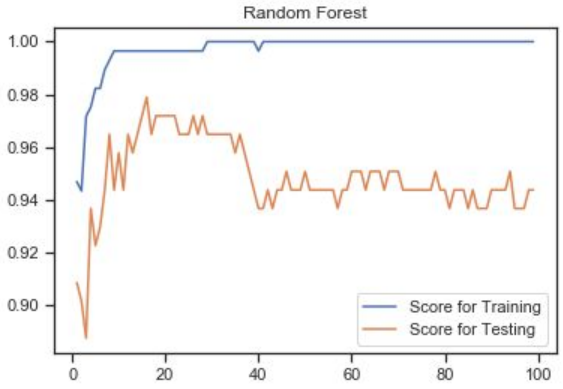


IV. TASK 4

A. *Random Forest*

Random Forest is applied to build classifiers on training data, and applied to predict the testing data with the number of estimators 1:100. The score chart is shown below in Figure 4.1. The best score is 97.9% with a number of estimator 16.

Figure 4.1 Score Chart of Random



B. *Rank Descriptive Features*

The rank of descriptive features are generated by implementing Random Forest with 10,000 tree trained. The ranks of descriptive features are listed in theTable 4.1.

Table 4.1 Ranks of Descriptive Features

Rank	Features	Importance%
1	perimeter3	0.15141193
2	area3	0.12493841
3	radius3	0.11284739
4	concave points3	0.10958672
5	concave points1	0.08298234
6	perimeter1	0.05108998
7	concavity3	0.04920028
8	area1	0.04815894
9	radius1	0.04076313
10	concavity1	0.03834779
11	area2	0.02748966
12	texture3	0.02041751
13	compactness3	0.02010525
14	texture1	0.01878105
15	compactness1	0.01260875
16	radius2	0.01142157
17	smoothness3	0.01129457
18	symmetry3	0.01069579
19	perimeter2	0.01003805
20	fractal dimension3	0.00768696
21	concavity2	0.00543757
22	smoothness1	0.00437981
23	symmetry1	0.00431045
24	symmetry2	0.00417979
25	compactness2	0.00402448
26	concave points2	0.00402064
27	fractal dimension2	0.00363733
28	fractal dimension1	0.0035655
29	smoothness2	0.00351525
30	texture2	0.0030631

C. *Conclusion*

Having the rank of features, we can select attributes that have higher importance. In this way, our models might be improved with better attributes selection.

It is fair to compare Random Forest with Decision Tree and KNN because they are compared on the same training and testing datasets, and the same model evaluation metrics (mean accuracy score).

Among all the algorithms, Random Forest has a better chance to win because it combines weak learners to build a more robust model that has a better generalization error and is less susceptible to overfitting. it has good classification performance, scalability and ease of use.

In order to implement a fair comparison test between different models, first I need to create a common training controller object, which means to have the same training and testing dataset. Second, I will apply the same model evaluation metrics on all models during the comparison.

V. TASK 5

A. *Drop Non Important Columns*

From Task 4, the importance ranks of descriptive attributes are generated. In this task, features with the importance% <0.005 are dropped. There are 20 descriptive attributes kept in the new datasets. All algorithms are re-run with new datasets, and the results are shown in Table5.1.

Table 5.1 Results Before and After Feature Dropping

Method	Best Score(before)	Best Score(After)
Decision Tree (Entropy)	94.3%	93.7%
Decision Tree (Gini)	90.8%	90.8%
KNN(Euclidean uniform weight)	97.9%	97.2%
KNN(Manhattan uniform weight)	98.6%	97.9%
KNN(Euclidean distance weight)	98.6%	97.2%
KNN(Manhattan distance weight)	98.6%	97.9%
Random Forest	97.9%	97.9%

The results after dropping non-important features show no significant improvement on each classifier.

B. *Generate New Features*

After dropping the non important features, the multicollinearity is analyzed. Their SPLOM plot is shown in Figure 5.1. The extremely highly correlated features (greater than $|\pm 0.9|$) are dropped with one most important feature kept. Then correlated features are aggregated meaningfully. There are 9 features left in the final model. The results before and after new feature generation are shown in Table 5.2.

In this test, the highest score 98.6% from part 2-4 is beaten by 99.3%, using a KNN classifier with Euclidean and uniform weight.

Fig 5.1 SPLOM Plot of Dataset with Important Attributes

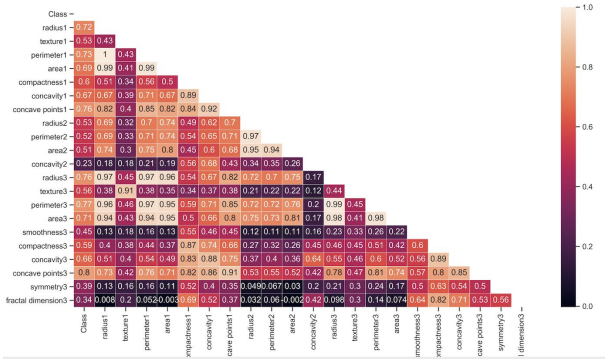


Table 5.1 Results Before and After Feature Dropping

Method	Best Score(Before)	Best Score(After)
Decision Tree (Entropy)	94.3%	94.4%
Decision Tree (Gini)	90.8%	93.7%
KNN(Euclidean uniform weight)	97.9%	99.3%
KNN(Manhattan uniform weight)	98.6%	97.2%
KNN(Euclidean distance weight)	98.6%	97.9%
KNN(Manhattan distance weight)	98.6%	97.2%
Random Forest	97.9%	97.9%

C. Solutions

As discussed above, the best score from task 2-4 is beaten. The results after dropping non-important features show no significant improvement on each classifier. However, after feature aggregation, KNN classifier with Euclidean and uniform weight achieves the highest accuracy score of 99.3%.

D. New Ideas

In our dataset, all the descriptive attributes are numeric. As we have already learned that one of the disadvantages of Decision Tree is that Decision Tree does not fit well for continuous variables. While working with continuous numerical variables, the decision tree loses information when it categorizes variables in different categories [3]. Considering the facts of Random Forest ensembles Decision Tree algorithm as well, I tried to apply Logistic Regression on the cancerData.

One of the assumptions of Logistic Regression requires little or no multicollinearity, so the similar technique as in task 5 is applied here as well. The final model has 8 descriptive attributes. The model evaluation metric is ROC curve, which is shown in Figure 5.2. And the confusion Matrix is displayed in Table 5.2.

Figure 5.2 ROC of Logistic Regression

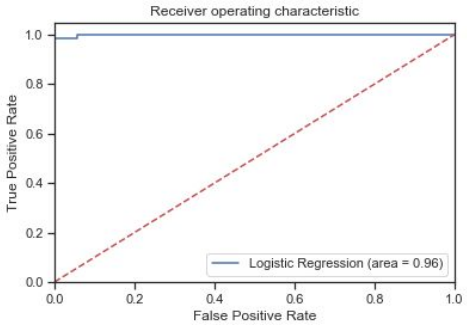


Table 5.2 Confusion Matrix

	Precision	Recall
0	0.99	0.94
1	0.95	0.99
Avg/Total	0.97	0.96

The Precision is 0.97. The Recall is 96%. Corresponding to our accuracy score, this Logistic Regression model has a score as high as 96%.

VI. REFERENCE

[1] Why Data Normalization is necessary for Machine Learning models

https://medium.com/@urvashilluniya/why-data-normalization-is-necessary-for-machine-learning-models-681b65a05029

[2] A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm

https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/

[3] What is a Decision Tree? How does it work?

https://clearpredictions.com/Home/DecisionTree