Exercise: Breast Cancer Prediction

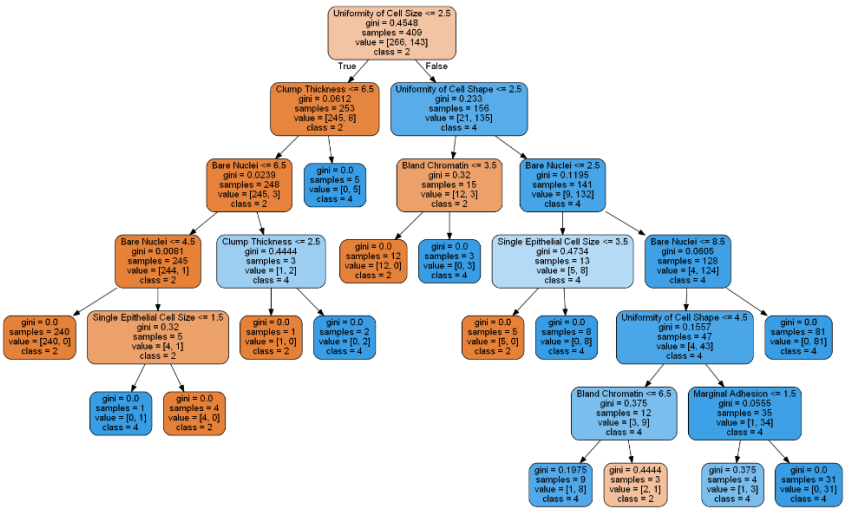
[Mining publicly available data] Download the dataset on breast cancer research from http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29. Specifically, click on the "Data Folder" link on the above page and download the following two files: (i) file "breast-cancer-wisconsin.data", which contains the actual data, and (ii) file "breast-cancer-wisconsin.names", which contains the description of the data. This dataset has 699 records, each record representing a different case of breast cancer. Each case is described with 11 attributes (indicated in the aforementioned .names file): attribute 1 represents case id, attributes 2-10 represent various physiological characteristics, and attribute 11 represents the type (benign or malignant). The dataset has several (16) records with missing values; you can delete these records before proceeding with the analysis.

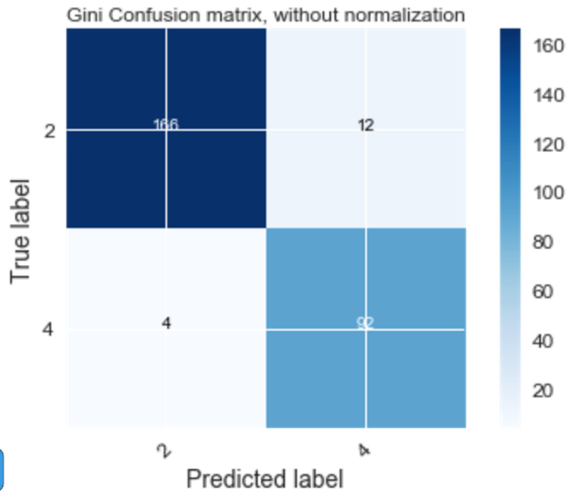
Use Python for this Exercise.

a) Perform a predictive modeling analysis on this same dataset using the decision tree, the k-NN technique and the logistic regression technique. Present a brief overview of your predictive modeling process, explorations, and discuss your results. Make sure you present information about the model "goodness" (possible things to think about: confusion matrix, predictive accuracy, precision, recall, f-measure).

Please refer to python notebook for steps.

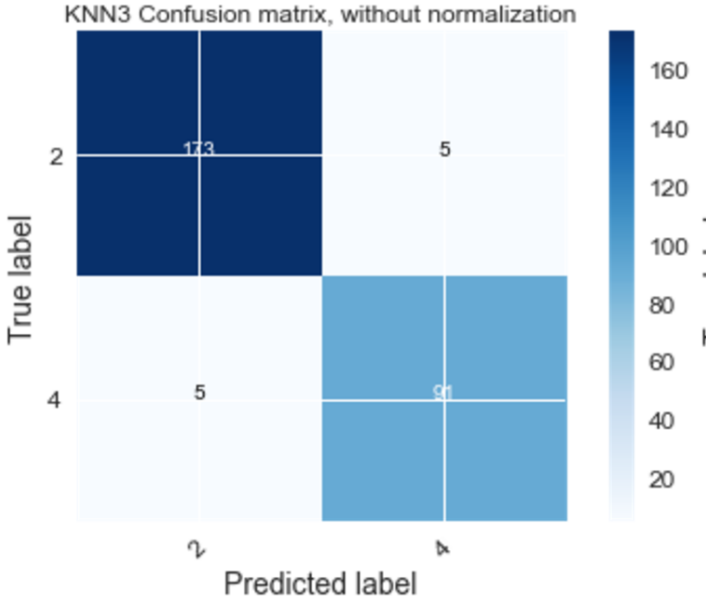
Decision Tree:

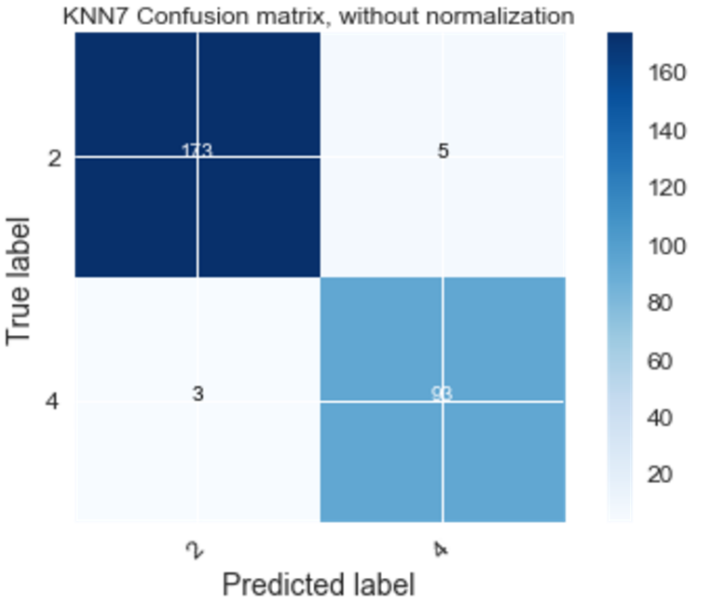





```
DT_Accuracy (out-of-sample): 0.94
DT_Accuracy (in-sample): 0.99
DT_F1 score (out-of-sample):  0.937011494253
DT_F1 score (in-sample)    :  0.991948660407
DT_Kappa score (out-of-sample):  0.874138723013
DT_Kappa score (in-sample)    :  0.983897426475
             precision    recall  f1-score   support

          2       0.98      0.93      0.95       178
          4       0.88      0.96      0.92        96

avg / total       0.94      0.94      0.94       274
```

KNN:

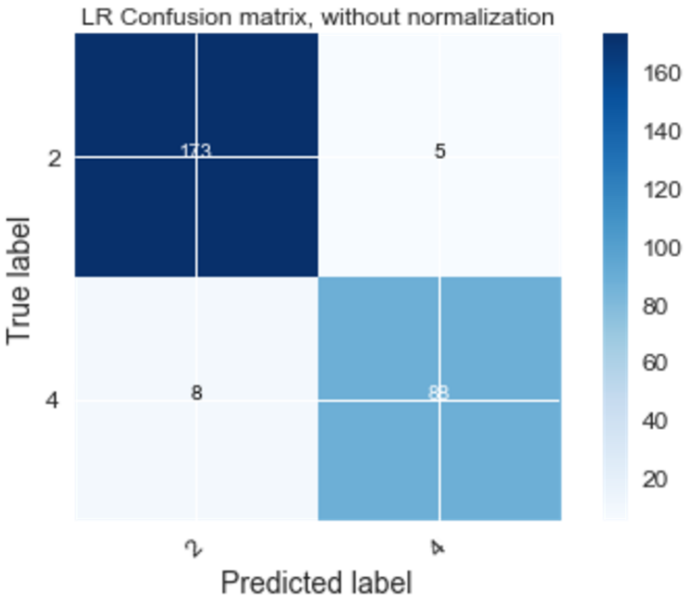





```
knn3_Accuracy (out-of-sample): 0.96
knn3_Accuracy (in-sample): 0.98
knn3_F1 score (out-of-sample):  0.959913389513
knn3_F1 score (in-sample)    :  0.981213540949
knn3_Kappa score (out-of-sample):  0.919826779026
knn3_Kappa score (in-sample)    :  0.962427328443
             precision    recall  f1-score   support

          2       0.97      0.97      0.97       178
          4       0.95      0.95      0.95        96

avg / total       0.96      0.96      0.96       274
```

```
knn7_Accuracy (out-of-sample): 0.97
knn7_Accuracy (in-sample): 0.98
knn7_F1 score (out-of-sample):  0.968082008271
knn7_F1 score (in-sample)    :  0.981152694118
knn7_Kappa score (out-of-sample):  0.936167734421
knn7_Kappa score (in-sample)    :  0.96230563638
             precision    recall  f1-score   support

          2       0.98      0.97      0.98       178
          4       0.95      0.97      0.96        96

avg / total       0.97      0.97      0.97       274
```

Logistic Regression:




```
LR_Accuracy (out-of-sample): 0.95
LR_Accuracy (in-sample): 0.98
LR_F1 score (out-of-sample):  0.947502616026
LR_F1 score (in-sample)    :  0.978495189022
LR_Kappa score (out-of-sample):  0.895019157088
LR_Kappa score (in-sample)    :  0.956990378043
             precision    recall  f1-score   support

          2       0.96      0.97      0.96       178
          4       0.95      0.92      0.93        96

avg / total       0.95      0.95      0.95       274
```

```
The weights of the attributes are: [[ 1.98  1.83 -0.72  0.89  0.26  2.04  0.9   0.19  0.94]]
```

Knn (k=7) has best out-of-sample performance (accuracy 97%)

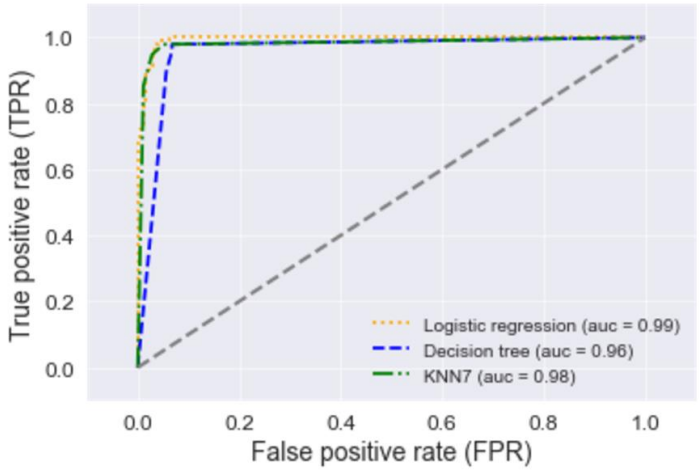
b) Explore how well k-NN performs for the following 2 distinct different parameter values k=3,7.

```
knn3_Accuracy (out-of-sample): 0.96
knn3_Accuracy (in-sample): 0.98
knn3_F1 score (out-of-sample):  0.959913389513
knn3_F1 score (in-sample)    :  0.981213540949
knn3_Kappa score (out-of-sample):  0.919826779026
knn3_Kappa score (in-sample)    :  0.962427328443
             precision    recall  f1-score   support

          2       0.97      0.97      0.97       178
          4       0.95      0.95      0.95        96

avg / total       0.96      0.96      0.96       274
```

```
knn7_Accuracy (out-of-sample): 0.97
knn7_Accuracy (in-sample): 0.98
knn7_F1 score (out-of-sample):  0.968082008271
knn7_F1 score (in-sample)    :  0.981152694118
knn7_Kappa score (out-of-sample):  0.936167734421
knn7_Kappa score (in-sample)    :  0.96230563638
             precision    recall  f1-score   support

          2       0.98      0.97      0.98       178
          4       0.95      0.97      0.96        96

avg / total       0.97      0.97      0.97       274
```

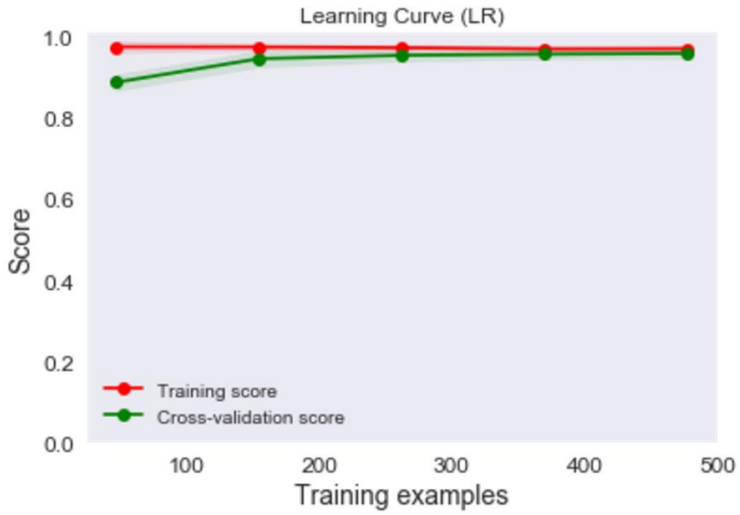
knn (k=7) has better out-of-sample performance

d) Create an ROC curve for k-NN, decision tree, and logistic regression. Discuss the results. Which classifier would you prefer to choose?



From ROC, we can know that **logistic regression** classifier is better (with higher AUC), it has lower False positive rate and higher True Positive rate (most northwest point).

e) Create a Learning Curve for the logistic regression technique.



From learning curve, we know that cross validation score improves as more training data become available. From 0 to 200, learning curve of cross validation score is steep as the modeling procedure finds more accurate model. However, after 200, learning curve becomes less steep and flattens out completely since 380 because the procedure no longer improve accuracy even with more training data.