〈생명정보학을 위한 데이터마이닝 project report〉

201801791 백다영

Motivation and contributions :

유방에서 실질조직 중 유선조직의 구성이 50%이상이면 치밀유방이라 부른다. 이는 유방촬영술로 하는 진단에 어려움을 주고 그 자체로 유방암 발생 확률을 높인다.

치밀유방은 해외에 비해 국내 여성들에서 더 높은 비중을 차지할 뿐만 아니라 유방촬영술을 넘어 비싼 유방초음파검사까지 필요로 한다.

그래서 나는 값싸고 간단한 미세침흡인세포검사(FNCA)의 중요성이 더욱 커질 것이라 생각해 FNCA 결과 분석에 관심을 갖게 되어 FNCA diagnostic dataset을 이용해서 Classification의 단일 모형인 decision tree와 이를 기반으로 앙상블을 한 boosting과 bagging을 이용하여 Comparative study/implementation /evaluation/extension of current mining methods을 진행하였다.

Related Works and methods : DecisionTree, RandomForest, XGBoost

Ensemble의 boosting 중 gradient descent boosting은 손실함수 error(Mi)를 미분해서 기울기가 0에 가까워지도록 가중치를 업데이트한다. 이를 기반으로 하여 오버피팅에 대한 규제함수와 병렬방향의 tree생성 등을 추가해 좀 더 개선된 모델이 바로 XGBoost이다.

Approach and methodology :

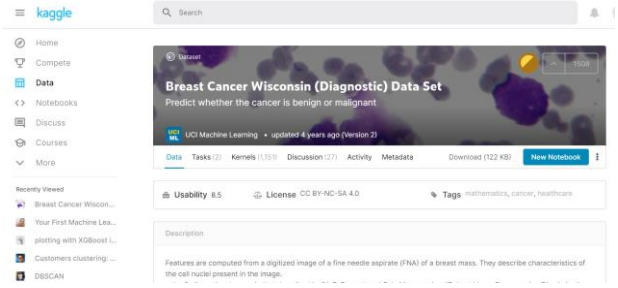
Data visualization으로 각 feature에 대한 클래스별 분포의 중앙값 위치 차이를 확인하였다. 이 차이가 클수록 그 특성이 분류에 중요할 것임을 예상할 수 있다.

Graphviz로 randomforest, decisiontree가 tree graph를 어떻게 만들었는지 보여주고 각 노드에 gini index를 나타내었다. 세 모델 모두 동일한 방법으로 아래와 같이 진행하였다.

Holdout method를 이용하여 데이터를 train : test = 9:1로 분리하고 accuracy, confusion matrix, classification report, ROC curve로 모델 성능 평가를 하였다. 데이터 수가 적어서 K-fold cross validation을 이용해 모델 평가를 하기 위해 한 fold(총 10개)마다 accuracy, confusion matrix, classification report, ROC curve가 나타나도록 하였다. 특성 중요도(feature importance)가 모델에 대한 이해와 데이터 분석에 도움을 줄 수 있을 것 같아 이것에 대한 코딩도 추가하였다. 예측이 왜 틀렸는지 왜 맞았는지를 분석할 수 있도록 한 fold당 높은 확률로 잘못 예측하거나 맞게 예측한 데이터의 특성들을 출력했다.

Evaluation and results :

https://www.kaggle.com/uciml/breast-cancer-wisconsin-data



1)ID number 2) Diagnosis (M = malignant, B = benign)

3~32)

a)radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

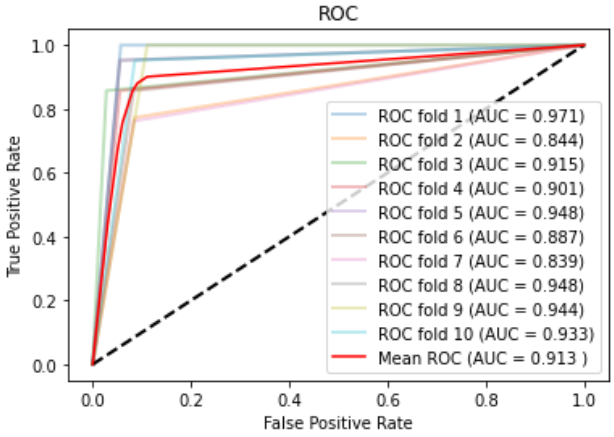
The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. All feature values are recoded with four significant digits. Class distribution: 357 benign, 212 malignant/ total 569

```
0.9649122807017544
[[33  2]
 [ 0 22]]
              precision    recall  f1-score   support

           0       1.00      0.94      0.97        35
           1       0.92      1.00      0.96        22

    accuracy                           0.96        57
   macro avg       0.96      0.97      0.96        57
weighted avg       0.97      0.96      0.97        57

[0.         0.02712804 0.         0.         0.00556724 0.00765496
 0.         0.         0.00626315 0.         0.00647727 0.
 0.         0.00234807 0.         0.         0.         0.
 0.00815749 0.02004208 0.7146971  0.03020372 0.00937641 0.02063155
 0.00819462 0.         0.0020352  0.13122309 0.         0.        ]
                        score
radius_worst          0.714697
concave points_worst  0.131223
texture_worst         0.030204
texture_mean          0.027128
area_worst            0.020632
fractal_dimension_se  0.020042
perimeter_worst       0.009376
smoothness_worst      0.008195
```



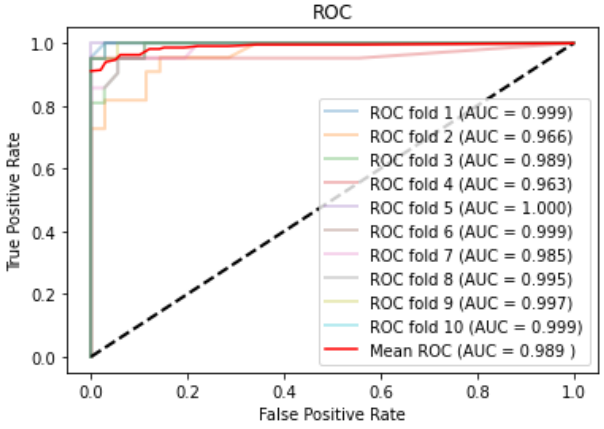
decisiontree

```
0.9824561403508771
[[35  1]
 [ 0 21]]
              precision    recall  f1-score   support

           0       1.00      0.97      0.99        36
           1       0.95      1.00      0.98        21

    accuracy                           0.98        57
   macro avg       0.98      0.99      0.98        57
weighted avg       0.98      0.98      0.98        57

[0.03866055 0.01143104 0.02647331 0.04663153 0.0083607  0.00766032
 0.08744124 0.09164022 0.00475329 0.00396207 0.01159512 0.00524397
 0.01758059 0.02768616 0.00448751 0.00400874 0.00578566 0.00869645
 0.00485671 0.00477003 0.11725554 0.01712627 0.08068152 0.13102378
 0.01143844 0.00937715 0.02899892 0.16946051 0.007472   0.00544066]
                         score
concave points_worst  0.169461
area_worst            0.131024
radius_worst          0.117256
concave points_mean   0.091640
concavity_mean        0.087441
perimeter_worst       0.080682
area_mean             0.046632
radius_mean           0.038661
```



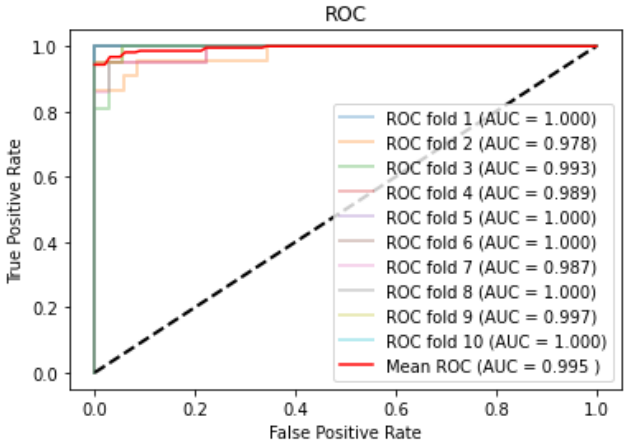
randomforest

```
1.0
[[35  0]
 [ 0 22]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        35
           1       1.00      1.00      1.00        22

    accuracy                           1.00        57
   macro avg       1.00      1.00      1.00        57
weighted avg       1.00      1.00      1.00        57

                       score
area_worst                55
texture_worst             54
area_se                   54
texture_mean              42
concave points_worst      38
concavity_worst           36
concave points_mean       24
smoothness_worst          20
```



XGBoost

각 모델에서 accuracy가 가장 높고 false negative(fn)의 수가 가장 낮은 fold의 결과를 사진으로 넣었다.

Decision tree의 모든 fold에서 radius_worst의 importance가 가장 높게 나왔다. Decision tree graph에서 radius_worst를 기점으로 불순도가 가장 많이 줄었기 때문이다. Data visualization에서 radius_worst의 클래스별 분포의 중앙값 위치 차이가 큰 것을 보면 이것이 feature importance와 연관이 있음을 확인할 수 있다. 한번도 쓰이지 않은 feature들도 있었는데 이는 decisiontree가 하나의 tree graph만 만들기 때문이다.

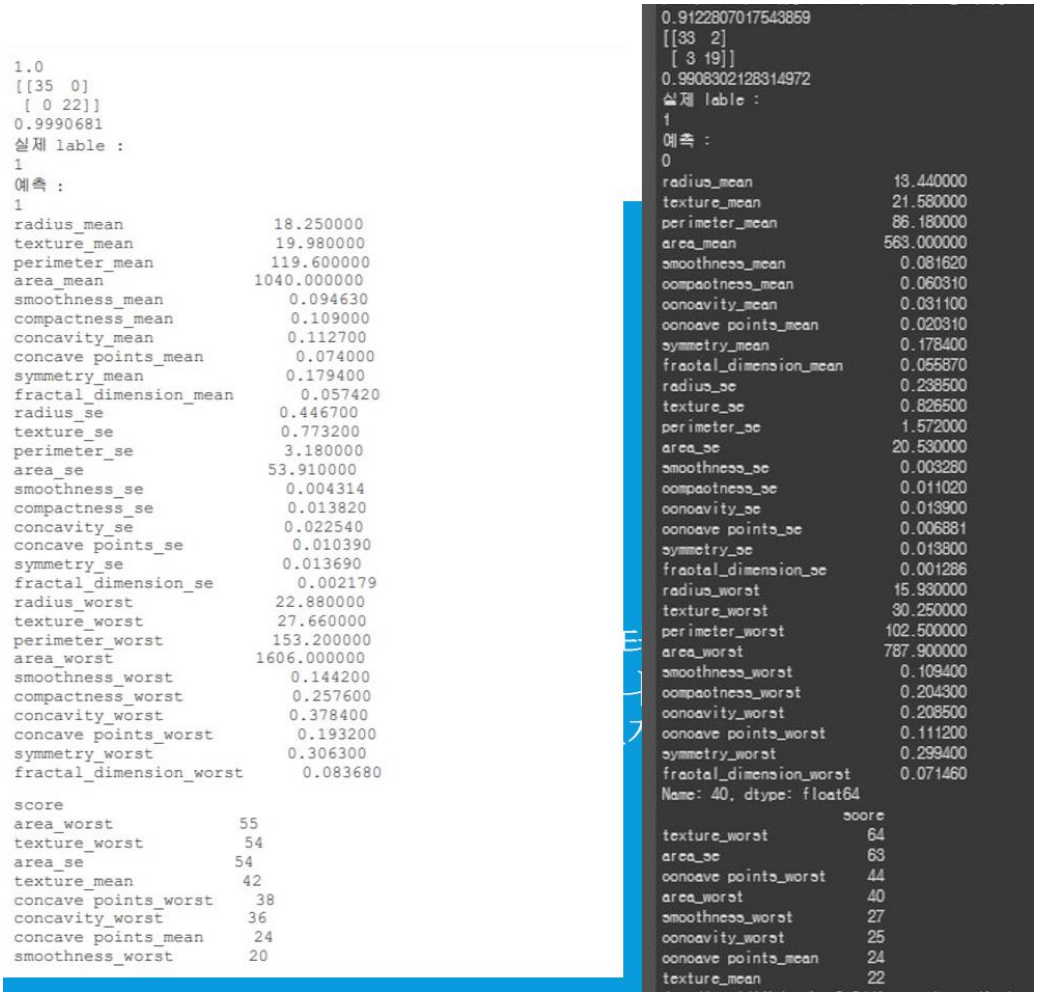
Randomforest는 100개(n_estimators)의 tree graph를 만들기에 모든 feature들을 사용하고 각 fold 별로 다른 feature importance 순위를 보였다. Randomforest 또한 불순도를 가장 많이 낮춘 특성을 더 중요하게 보기 때문에 주로 data visualization에서 클래스별 분포의 중앙값 위치 차이가 큰 feature들이 importance가 높게 나온 것을 확인할 수 있었다.

XGBoost는 얼마나 트리에 자주 나왔는지, 즉, splits point로 자주 나온 feature순으로 importance를 측정했다.

K-fold cross validation에서 평균적으로 decisiontree는 2.2, randomforest는 1.8, XGBoost는 1.4의 fn수를 보였기에 환자를 놓치지 않는 것이 더 중요한 medical 관점에서는 ensemble이 decisiontree보다 더 좋은 예측을 했음을 알 수 있다. 만약 fn의 수가 세 모델 모두 3으로 나온 holdout method에만 머물렀다면, ensemble의 강점을 알지 못했을 것이다. 그리고 세 모델 모두 holdout method에 비해 k-fold cross validation에서 AUC가 더 크게 나온 것을 볼 수 있다. K-fold cross validation으로 결과의 신뢰성을 높임과 동시에 평균적으로 모델의 학습 또한 더 잘되었음을 알 수 있다.

Conclusion and future work :

a) 나는 이번 프로젝트를 통해 boosting 기법 중 하나인 XGBoost를 새롭게 알게 되었다. XGBoost가 최근 대회에서 많은 우승을 차지했다는 점이 흥미로워 이를 이용해 새로운 모델을 만들어 보았다. testset에 대해 가장 높은 AUC, 평균적으로 제일 낮은 fn수를 보여주며 XGBoost의 유용함을 알 수 있었다. 결론적으로, 새로운 기법을 찾았다면 원하는 결과가 나오도록 그것을 데이터에 잘 적용시키는 것이 데이터 마이닝에서 정말 중요한 것임을 알게 되었다.



b)

위 사진은 임의의 한 fold에서 XGBoost가 높은 확률로 실제 값과 동일하게 예측한 것과 다르게 예측한 것의 각 feature value를 출력한 사진이다.

만약 머신러닝이 없었다면, 사람이 trainset 데이터의 30개 feature들을 일일이 확인하면서 testset의 결과를 예측해야 했을 것이다. 하지만 모델을 학습시킨 후 testset 검증에서 극적인 차이를 보이는 예들을 통해서 우리는 결과에 대한 feature 분석을 더 쉽고 간단하게 할 수 있다.

현재는 기계가 직접 이미지에서 특징을 추출해 분류를 하는 단계까지 이르렀다. 허나, 나의 프로젝트를 통해 기계가 어느 feature에 어떻게 집중하는지를 알 수 있다고 생각한다. 그로 인해 사람이 기계가 데이터를 해석하는 관점까지 파악함으로써 더 넓고 다양한 시각에서 데이터를 분석할 수 있을 것이라 생각한다.