NAVER DATA SCIENCE COMPETITION

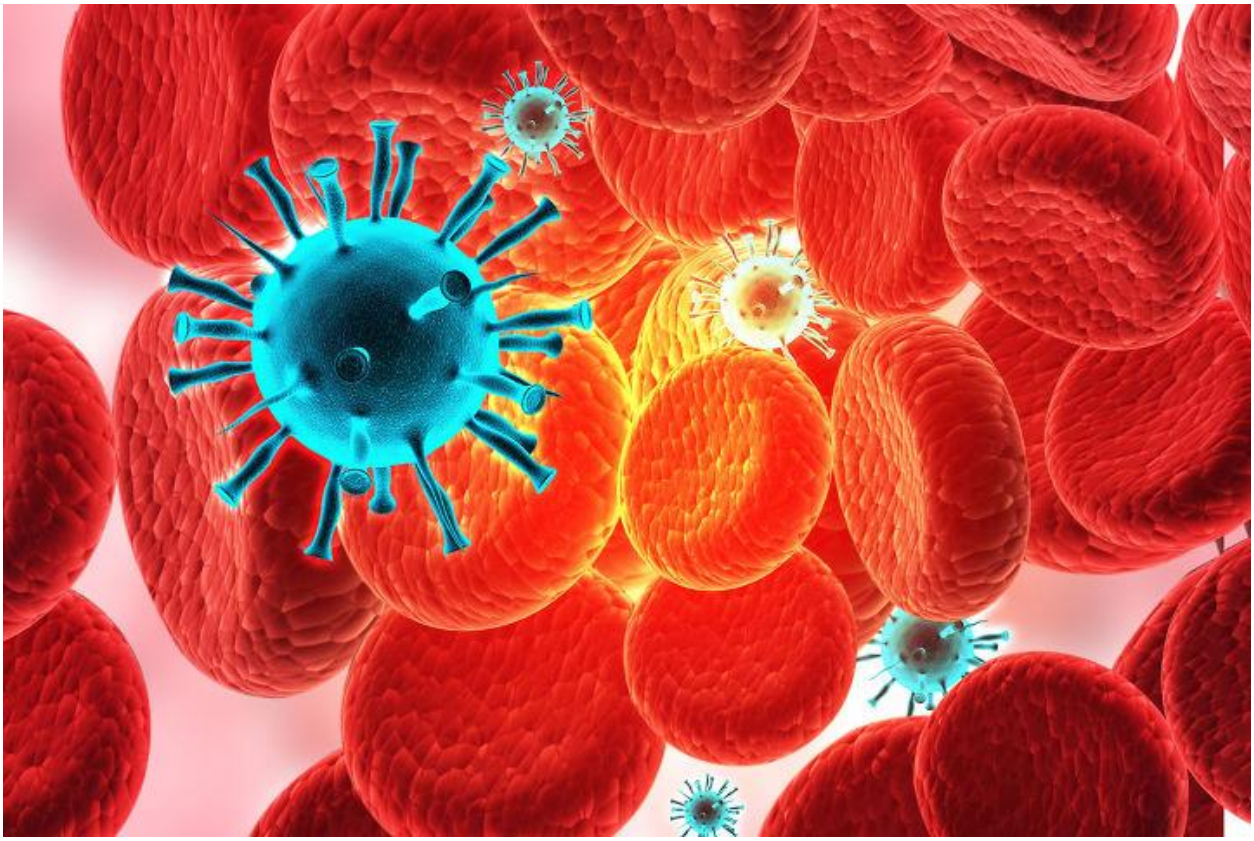
WISCONSIN DIAGNOSTIC BREAST CANCER DATA SET

팀장: 이재융(19931001)

메일: jaeyung1001@naver.com

팀 구성원: 이재융(19931001) / 박해주(19940225)

"현생 인류의 영원한 숙제"



암이란 인류가 가장 무서워하고 가까이 있는 질병이다. 현재 암에 의한 사망 수는 타 질병들과 비교해 봐도 상위권이고 현대인의 환경 역시 암이 생기기 쉬운 상황이라 계속 증가추세에 있다. 또한 발생하고 조기발견 및 치료가 이루어지지 않으면 병기가 진행되면서 전이와 함께 사망률이 크게 증가하는 반면 비교적 완만히 죽어간다는 것이다. 하지만 암은 불치병이 아니다, 조기에 발견했을시 암을 이겨낼수 있으며 약 70%이상이 완치가 가능하다. 하지만 모든 일반인들이 의사가 아니기때문에 암을 발견하기가 쉽지가 않다. 따라서 우리는 유방 암에대한 여러가지 특징점들과 기계학습을 이용하여 일반인들도 쉽게 암을 진단할 수가 있는 솔루션을 제안한다.

DATA SET EXPLAINATION

- 데이터 출저 : *https://www.kaggle.com/uciml/breast-cancer-wisconsin-data#data.csv*

- 악성 암 환자 수: 212 명, 양성 암 환자 수: 357 명의 유방암에 관한 데이터
- 총 33 가지 feature data set

Feature Name	Explaination
Id	Patient id
Diagnosis	Whether Patient's cancer is malignant or benign
Radius	Distances from center to points on the perimeter
Texture	Standard deviation of gray-scale values
Perimeter	Size of the core tumor
Area	Cancer area
Smoothness	Local variation in radius lengths
Compactness	Perimeter^2 / area - 1.0
Concavity	Severity of concave portions of the contour
Concave points	Number of concave portions of the contour
Symmetry	Same as data name
[]_mean	Data value mean
[]_se	Data value standard error
[]_worst	Three max value mean from data
Fractal_dimension	Mean for "coastline approximation" - 1
Unnamed: 32	Nothing

ENVIRONMENT

개발환경은 다음과 같다

- OS: Windows 10
- Language : Python
- Github URL: https://github.com/jaeyung1001/naver_competition

DATA ANALYSIS

저희는 주어진 csv 파일에 대해서 안에 어떤 데이터가 들어있고 또 어떠한 데이터형식으로 저장되어있는지 확인해보았다. 결과는 다음 그림 1 과 같다.

```
In [2]:  1 data = pd.read_csv('data.csv')
         2 data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
id                         569 non-null int64
diagnosis                  569 non-null object
radius_mean                569 non-null float64
texture_mean               569 non-null float64
perimeter_mean             569 non-null float64
area_mean                  569 non-null float64
smoothness_mean            569 non-null float64
compactness_mean           569 non-null float64
concavity_mean             569 non-null float64
concave points_mean        569 non-null float64
symmetry_mean              569 non-null float64
fractal_dimension_mean     569 non-null float64
radius_se                  569 non-null float64
texture_se                 569 non-null float64
perimeter_se               569 non-null float64
area_se                    569 non-null float64
smoothness_se              569 non-null float64
compactness_se             569 non-null float64
concavity_se               569 non-null float64
concave points_se          569 non-null float64
symmetry_se                569 non-null float64
fractal_dimension_se       569 non-null float64
radius_worst               569 non-null float64
texture_worst              569 non-null float64
perimeter_worst            569 non-null float64
area_worst                 569 non-null float64
smoothness_worst           569 non-null float64
compactness_worst          569 non-null float64
concavity_worst            569 non-null float64
concave points_worst       569 non-null float64
symmetry_worst             569 non-null float64
fractal_dimension_worst    569 non-null float64
Unnamed: 32                0 non-null float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

Figure 1. Data description

Id 와 diagnosis 를 제외한 모든 데이터들이 float64 형식으로 저장되어있었고 총 569 줄의 데이터가 저장되어 있었다. 그림 2 와 같이 569 개의 데이터 중 악성 유방암 환자 데이터 수는 212 개, 양성 유방암 환자 데이터 수는 357 개가 있다.

암환자 데이터 수는: 212개, 정상환자 데이터 수는 357개

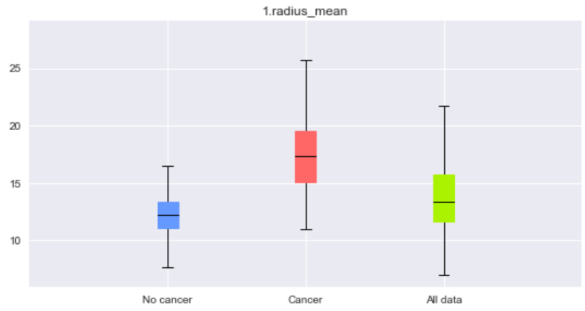
```
In [8]:  1 cancer_data = data[data['M']== 1]
         2 nocancer_data = data[data['M'] == 0]
```

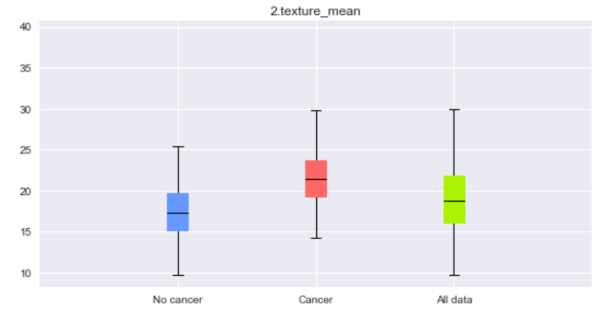
```
In [42]:  1 print("the malignant cancer data: {}, benign cancer data: {}".format(len(cancer_data), len(nocancer_data)))
```

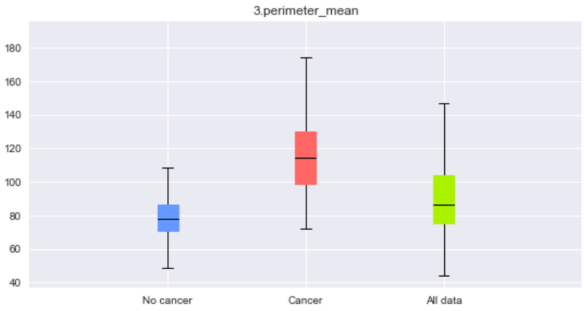
```
the malignant cancer data: 212, benign cancer data: 357
```

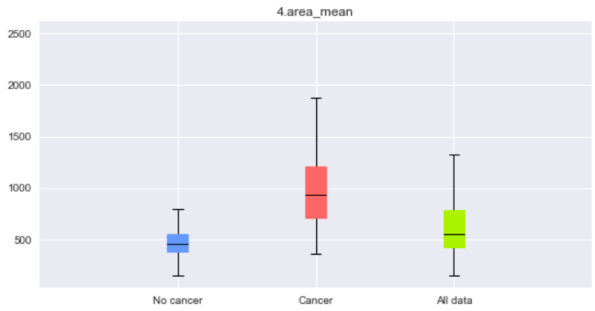
Figure 2. Count M/B cancer data

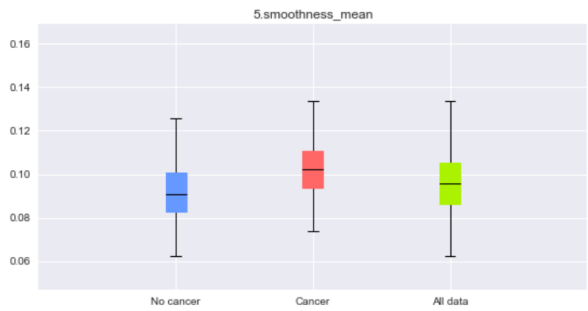
또한 저희는 그림 3 과 같이 각 데이터들의 boxplot 을 그리고 조사를 진행해 보았다.

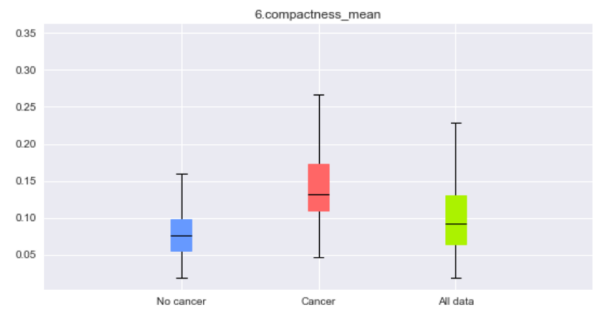

1.radius_mean
25
20
15
10
No cancer
Cancer
All data

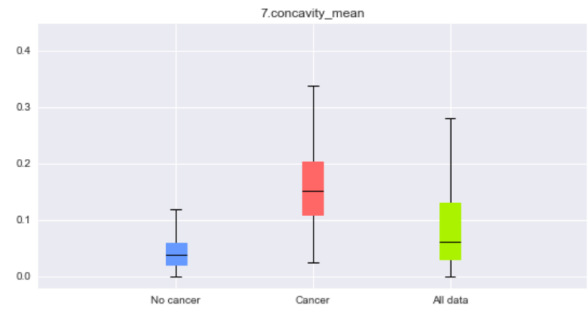

2.texture_mean
40
35
30
25
20
15
10
No cancer
Cancer
All data

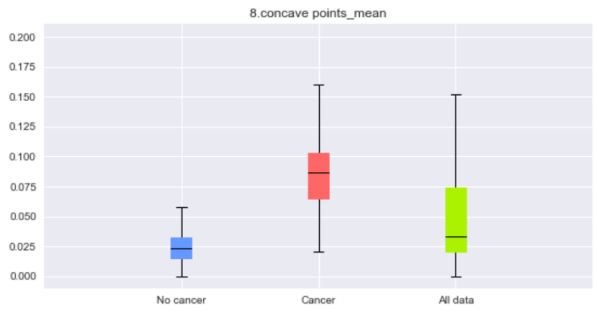

3.perimeter_mean
180
160
140
120
100
80
60
40
No cancer
Cancer
All data

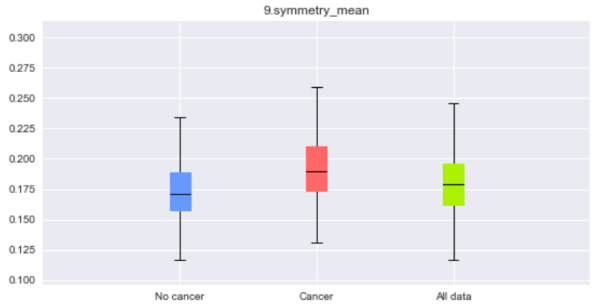

4.area_mean
2500
2000
1500
1000
500
No cancer
Cancer
All data

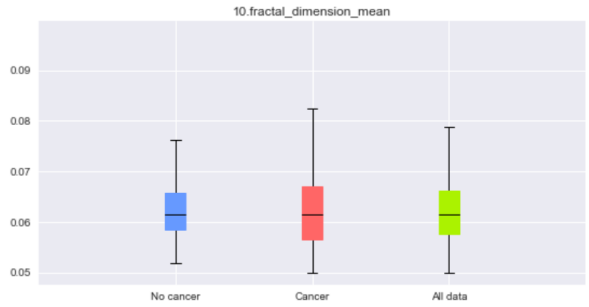

5.smoothness_mean
0.16
0.14
0.12
0.10
0.08
0.06
No cancer
Cancer
All data

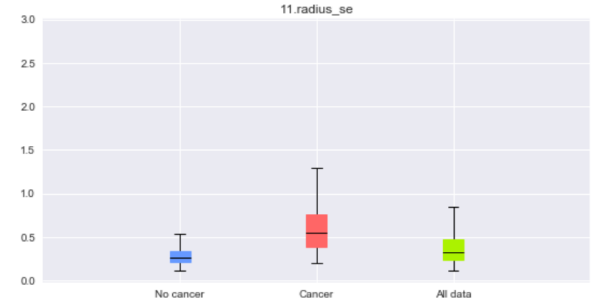

6.compactness_mean
0.35
0.30
0.25
0.20
0.15
0.10
0.05
No cancer
Cancer
All data

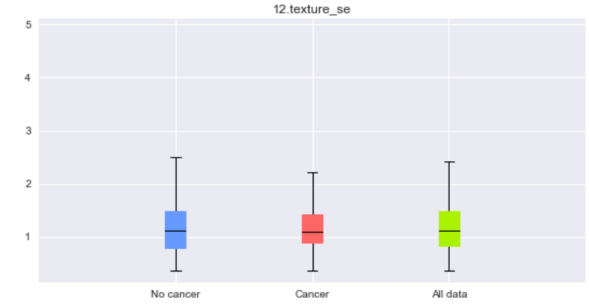

7.concavity_mean
0.4
0.3
0.2
0.1
0.0
No cancer
Cancer
All data

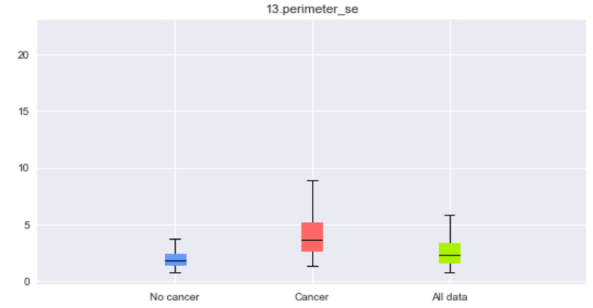

8.concave points_mean
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
No cancer
Cancer
All data

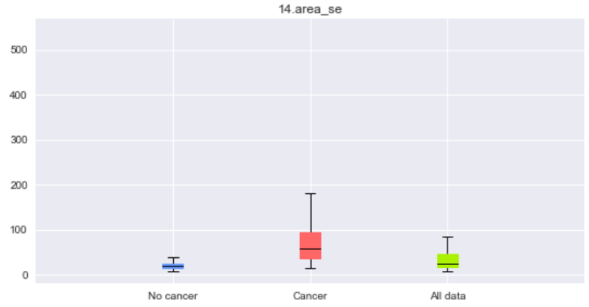

9.symmetry_mean
0.300
0.275
0.250
0.225
0.200
0.175
0.150
0.125
0.100
No cancer
Cancer
All data

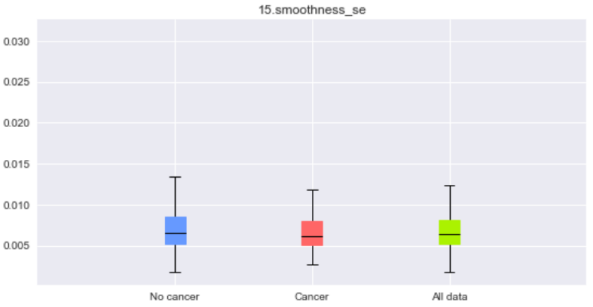

10.fractal_dimension_mean
0.09
0.08
0.07
0.06
0.05
No cancer
Cancer
All data

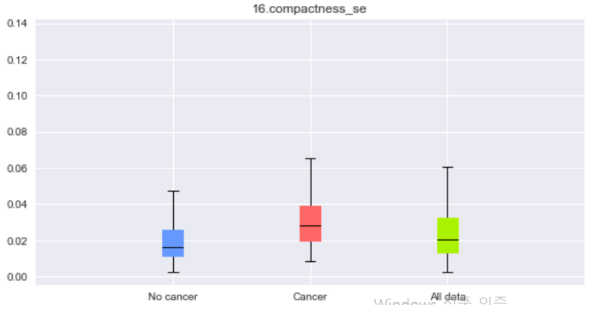

11.radius_se
3.0
2.5
2.0
1.5
1.0
0.5
0.0
No cancer
Cancer
All data

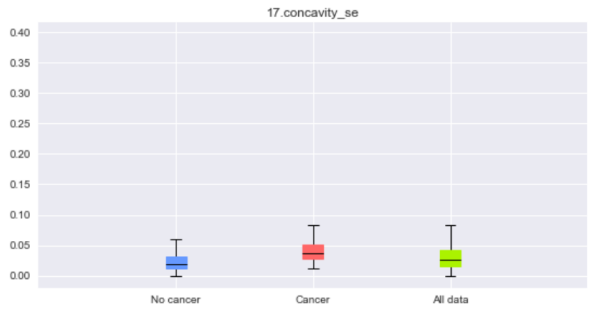

12.texture_se
5
4
3
2
1
No cancer
Cancer
All data

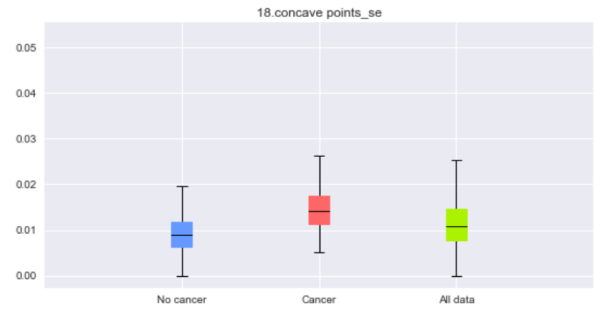

13.perimeter_se
20
15
10
5
0
No cancer
Cancer
All data

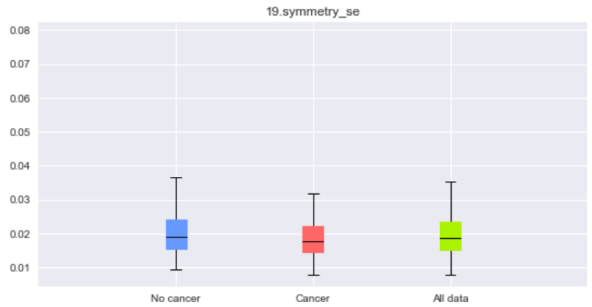

14.area_se
500
400
300
200
100
0
No cancer
Cancer
All data

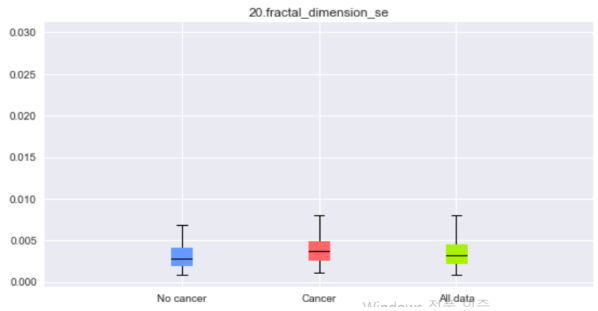

15.smoothness_se
0.030
0.025
0.020
0.015
0.010
0.005
No cancer
Cancer
All data

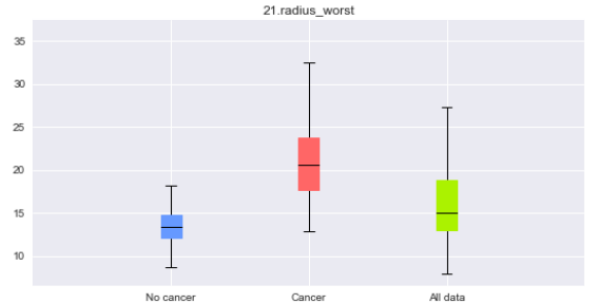

16.compactness_se
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
No cancer
Cancer
All data

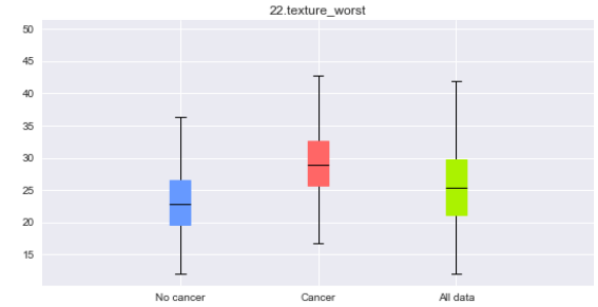

17.concavity_se
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
No cancer
Cancer
All data

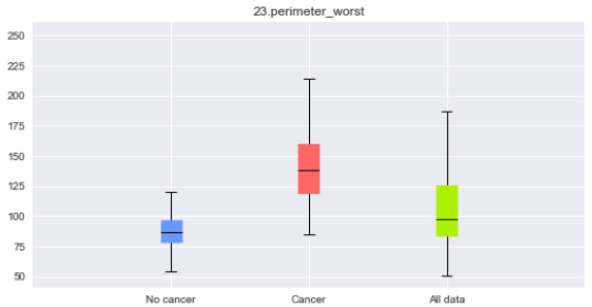

18.concave points_se
0.05
0.04
0.03
0.02
0.01
0.00
No cancer
Cancer
All data

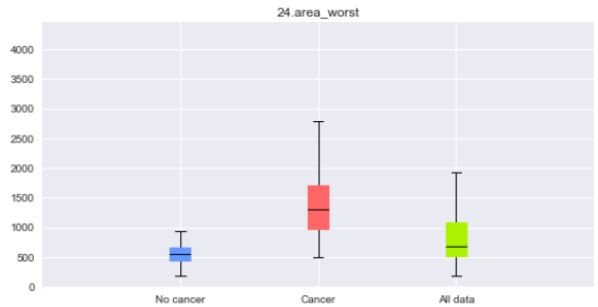

19.symmetry_se
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
No cancer
Cancer
All data

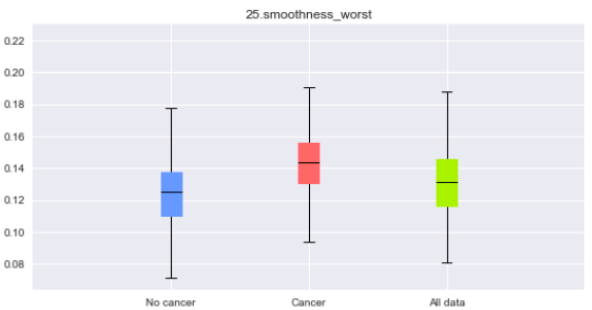

20.fractal_dimension_se
0.030
0.025
0.020
0.015
0.010
0.005
0.000
No cancer
Cancer
All data

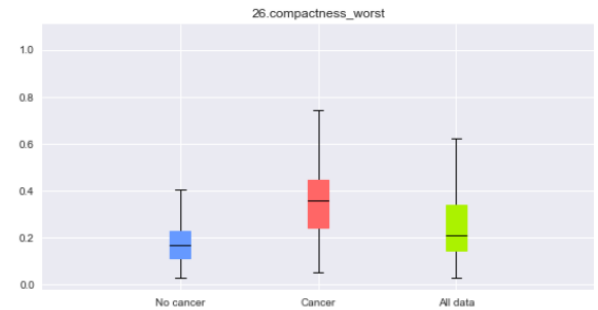

21.radius_worst
35
30
25
20
15
10
No cancer
Cancer
All data


22.texture_worst
50
45
40
35
30
25
20
15
No cancer
Cancer
All data


23.perimeter_worst
250
225
200
175
150
125
100
75
50
No cancer
Cancer
All data


24.area_worst
4000
3500
3000
2500
2000
1500
1000
500
0
No cancer
Cancer
All data


25.smoothness_worst
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
No cancer
Cancer
All data


26.compactness_worst
1.0
0.8
0.6
0.4
0.2
0.0
No cancer
Cancer
All data

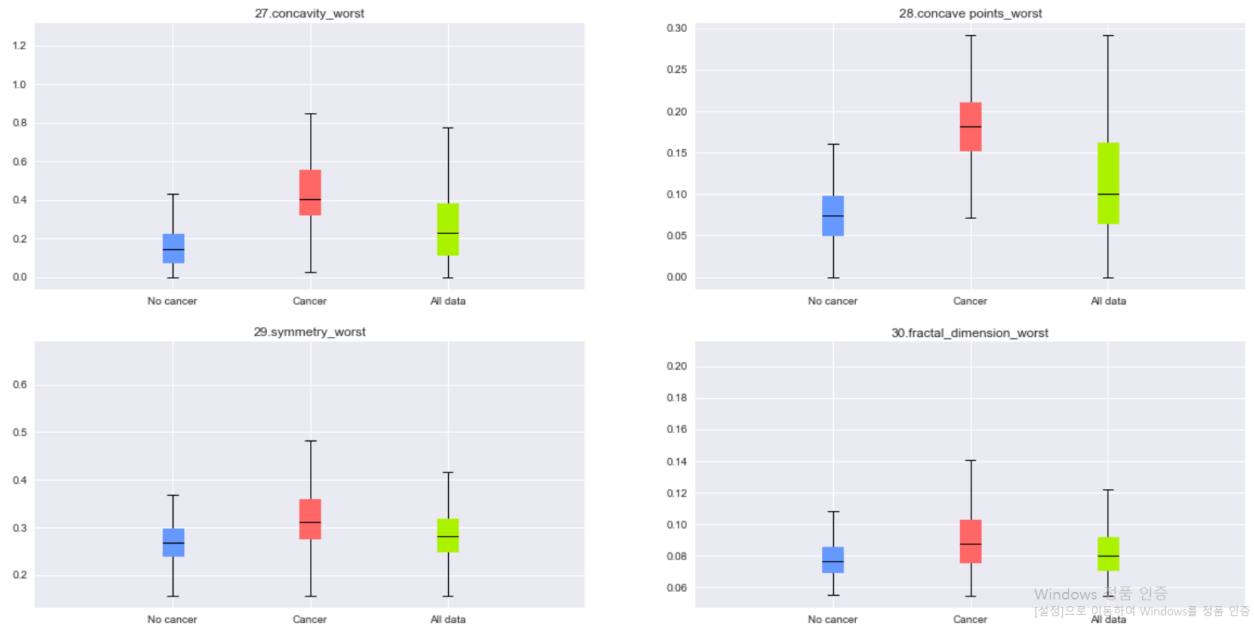


Figure 3.Each data set boxplot

그림 3 의 boxplot 을 조사하면, 악성과 양성 암환자 데이터에 별로 차이가 없는 데이터그림을 볼수가 있다(ex. Symmetry_se, Texture_se, fractal_dimension_se 등). 우리는 이러한 데이터가 악성 암 판단 알고리즘에 불필요한 정보(정확도에 영향을 주는 데이터)라고 가정을 하고 악성&양성 암 판별과 다른 데이터 셋들의 pearson correlation coefficient 값들을 구해보았다. 결과는 다음과 같다.

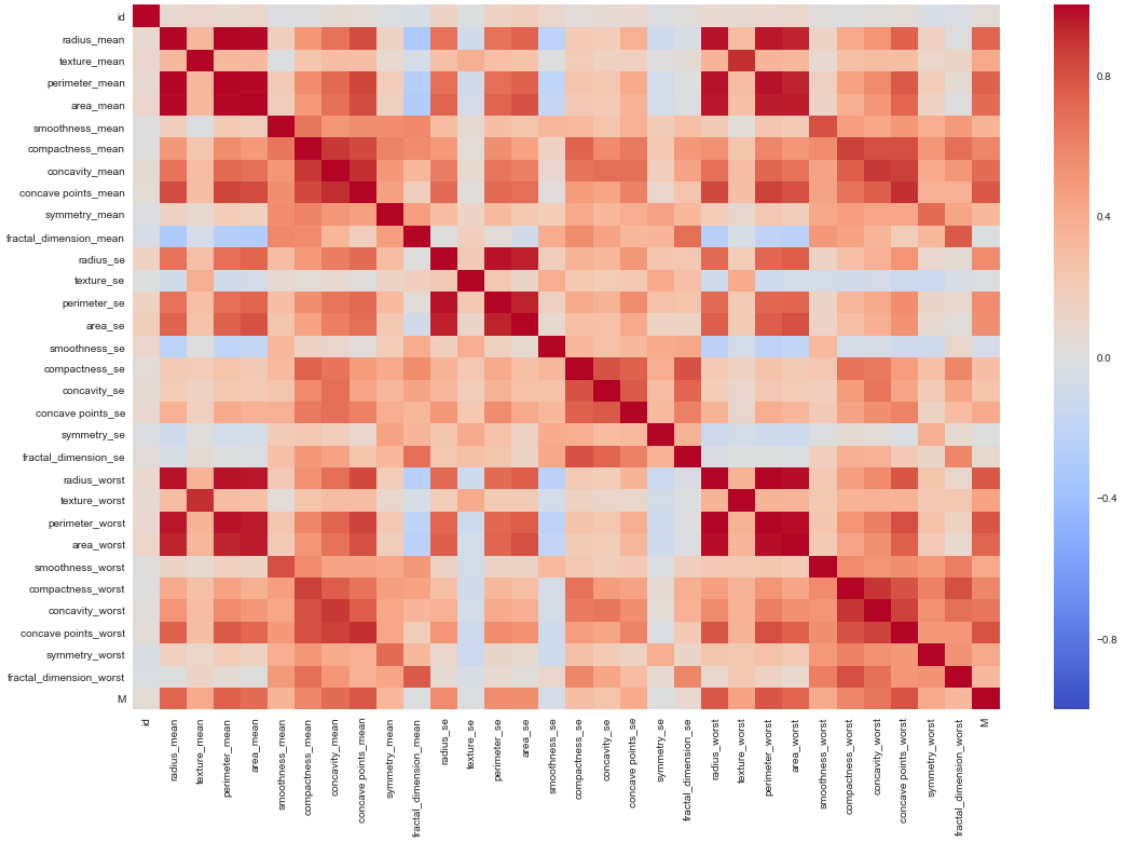


Figure 4. Correlation

모든 데이터의 상관계수 값이 중요하지만 이번 분석에서 중요하게 봐야할것은 M 열에 있는 수치 값이다. M 열에 있는 수치 값을 살펴서 상관계수가 낮은 데이터를 삭제한다. 우리는 진행 당시 texture_se, texture_mean, texture_worst, symmetry_se, smoothness_se 데이터 값을 삭제하고 기계학습 알고리즘에 학습 시켰다.

TRAINING MODEL

악성 유방암 판단 여부 알고리즘은 총 세가지를 썻고, 딥러닝 알고리즘은 한가지를 적용해보았다.

- 기계학습 알고리즘
  - Logistic Regression
  - Decision Tree
  - Random Forest
- 딥러닝 알고리즘
  - Deep Neural Network

그림 5 는 우리가 적용한 기계학습 알고리즘중 하나인 Decision Tree 의 그래프를 그려본것이다.

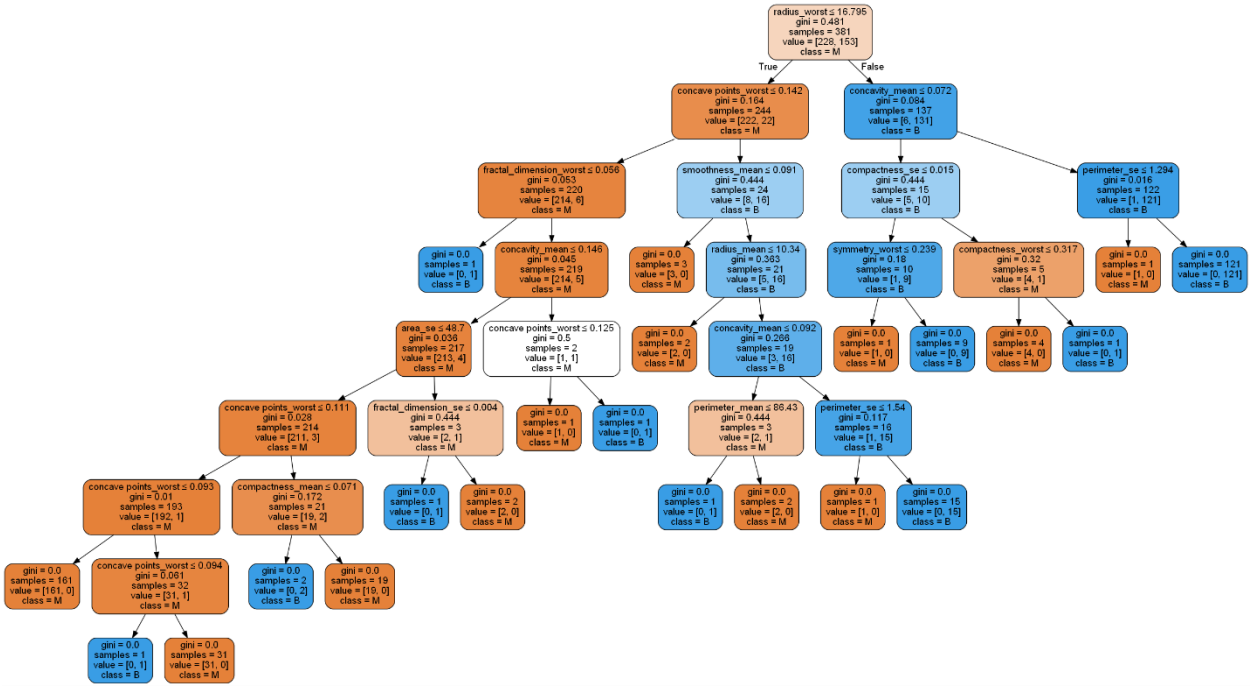
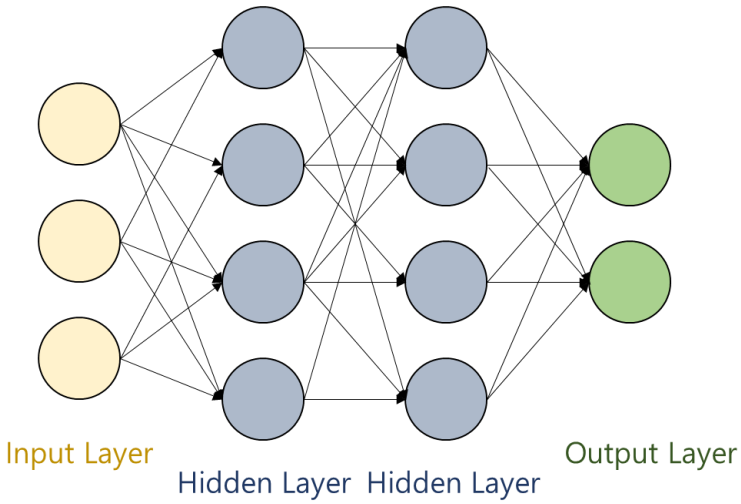


Figure 5.Decision Tree graph

또, 다음과 같은 간단한 MLP 네트워크로도 성능을 확인해 보았다.



기존 머신러닝 알고리즘과 달리 MLP 에서는 제외되는 feature 없이 모든 feature 들을 다 사용 하여 예측을 했고 각 layer 의 parameter 들은 아래와 같다.

Input Layer: 30

Hidden Layer: 100

Activation function = Relu

Dropout rate = 0.1

Drop out 과 batch normalization 는 FC-layer 에 적용 되었다.

EXPERIMENTAL RESULTS

결과적으로 네 가지 알고리즘의 결과는 다음 표와 같다. 우리는 학습데이터와 평가데이터를 랜덤으로 0.66:0.33 로 나누어 평가를 하였고, 정확도는 아래 표와 같다.

```
In [78]:
1 print("LogisticRegression accuracy: {}%".format((precision_score(Y_test, predicted_lr) * 100)))
2 print("DecisionTree accuracy: {}%".format((precision_score(Y_test, predicted_dt) * 100)))
3 print("RandomForest accuracy: {}%".format((precision_score(Y_test, predicted_rf) * 100)))

LogisticRegression accuracy: 91.80327868852459%
DecisionTree accuracy: 90.32258064516128%
RandomForest accuracy: 98.18181818181819%
```

Algorithm	Accuracy
LogisticRegression	92%
Decision Tree	88%~90%
Random Forest	92%~98%
Deep Neural Network	96%~98%

가장 높은정확도를 보여준 Random Forest 과 DNN 알고리즘은 98%의 정확도를 보여주었으며, Decision Tree 의 정확도가 제일 낮게 나왔다.

우리는 분석 도중 다음과 같은 생각이 들었다. **"이 정도면 엄청 높은 정확도인데 왜 그럼에도 불구하고 많은 사람들이 암으로 죽어 나갈까? "** 여기엔 여러 가지 요인이 있을 수 있는데 그 중 하나는 데이터에 있었다. **"실생활의 데이터도 과연 악성 암과 양성 암의 데이터가 5:5 비율로 존재를 할까?"** 즉 실 생활의 데이터의 비율은 5:5 가 아닌 한쪽 편에 치우쳐져 있을 것이란 것이다. 이를 데이터 불균형 (Data imbalance) 문제라고 하며 현재 데이터 사이언스가 접목된 많은 분야에서 관심을 받고 있다. 우리는 이 문제를 살펴보기 위해 다음과 같은 간단한 실험을 해보았다.

1. 악성 암데이터 : 양성 암데이터 비율 ➔ 6:4
2. 악성 암데이터 : 양성 암데이터 비율 ➔ 7:3

다음과 같이 데이터 중 양성과 악성의 비율을 조절하여 detection 실험을 진행해 보았으며 결과는 다음과 같다.

Decision Tree 알고리즘 그렇게 큰 변화를 보여주지 않았고, Logistic Regresson 과 Random Forest 알고리즘은 악성 양성 데이터의 비율에 따라서 정확도가 떨어지는 것을 확인 할 수가 있었고 이는 데이터 균형이 머신러닝 알고리즘의 결과에 영향을 미친 다는 것을 의미한다.