Data Mining

의사 결정 트리 과제

2021년 5월 20일

유방암 진단

의사 결정 트리로 유방암을 진단해보자.

kNN

accuracy : 0.956
precision : 0.97
recall : 0.746
f1_score : 0.843

로지스틱 회귀

accuracy : 0.965
precision : 1.0
recall : 0.901
f1_score : 0.948

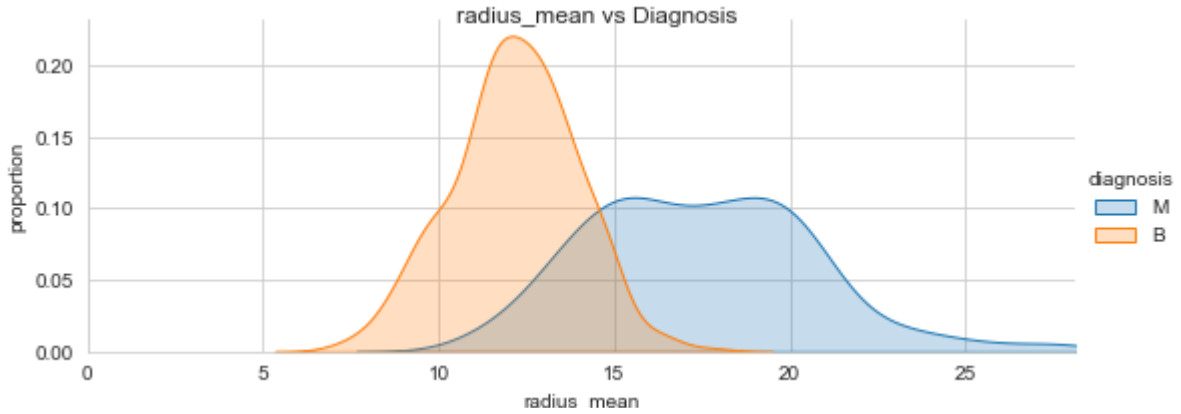
의사 결정 트리

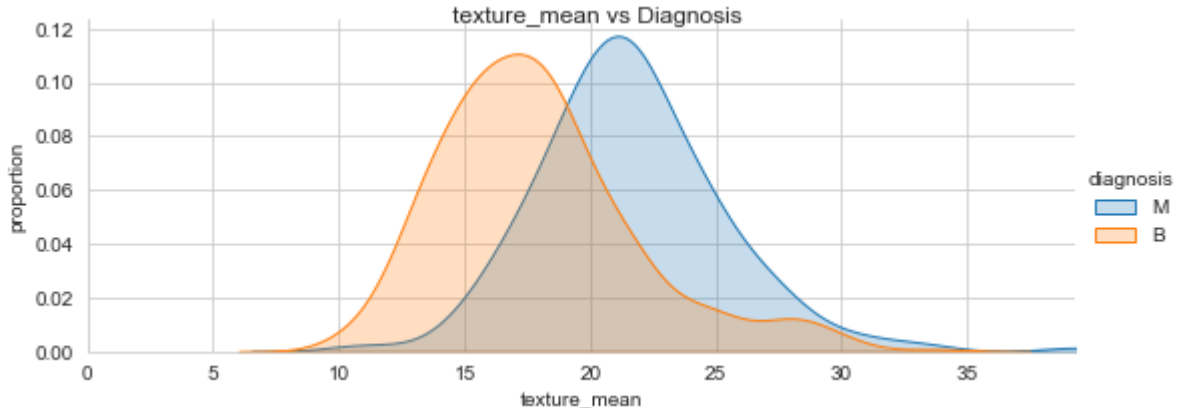
accuracy : 0.97

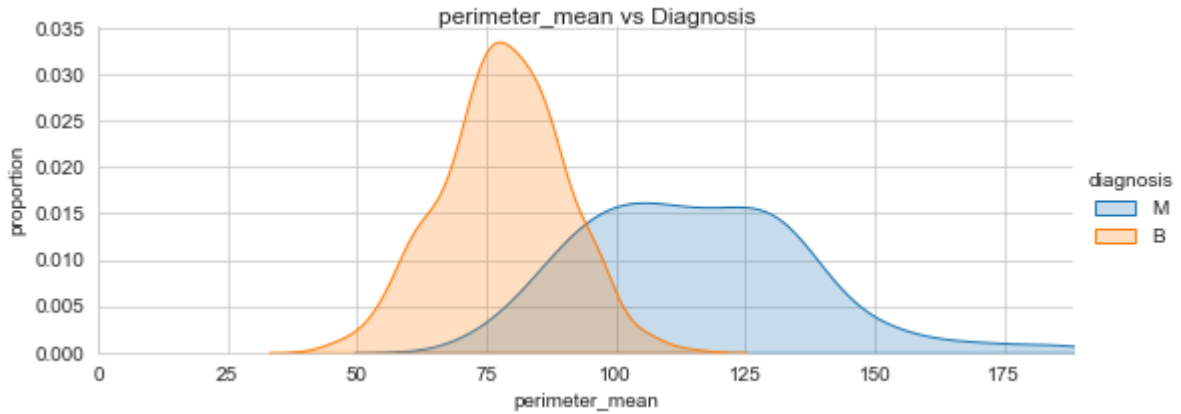
랜덤 포레스트

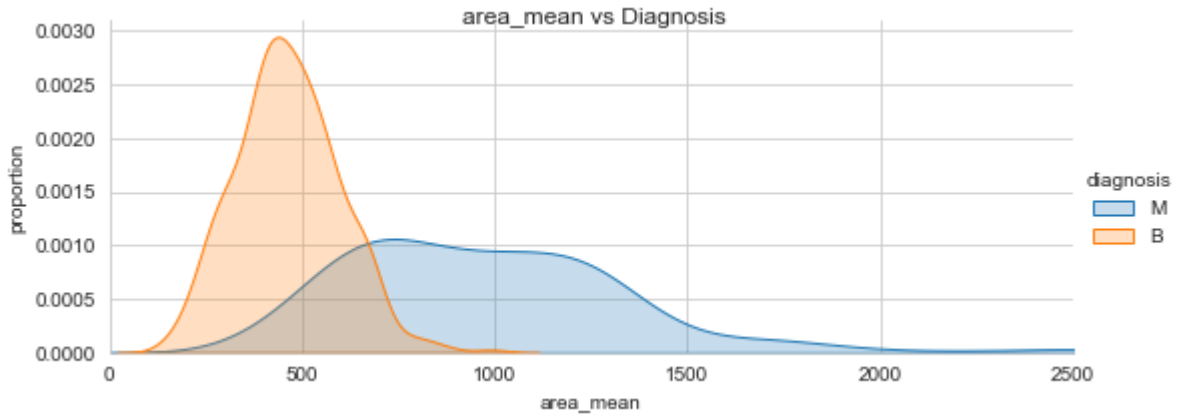
데이터 탐색 특징 별 분포

특징 30개 for loop 문으로 돼있음

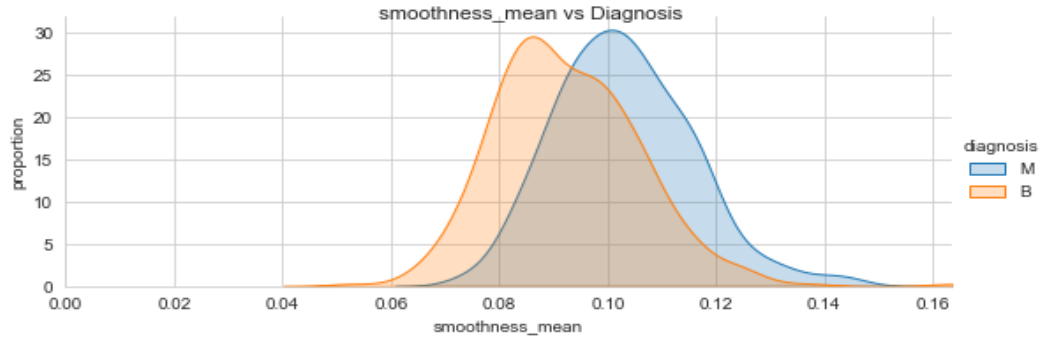


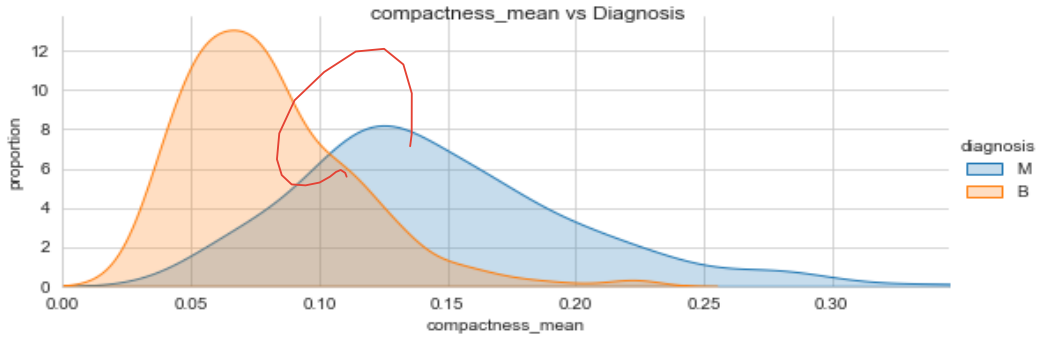


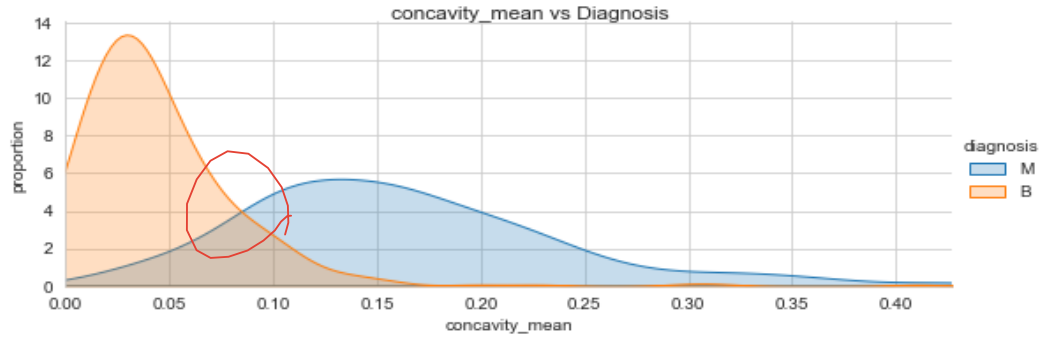


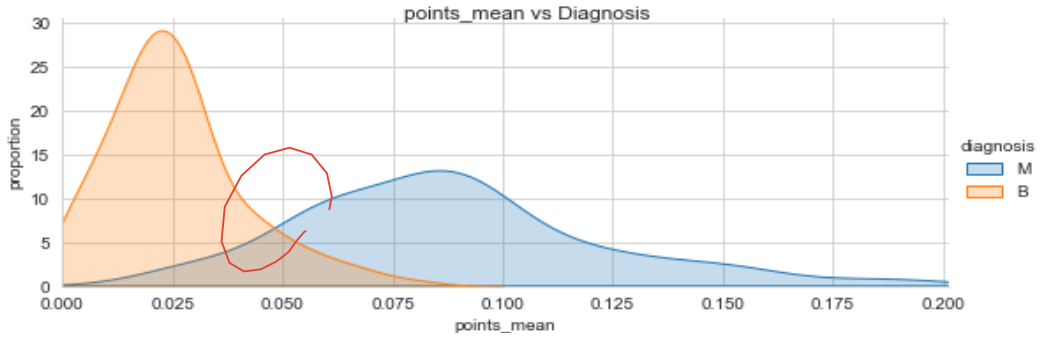


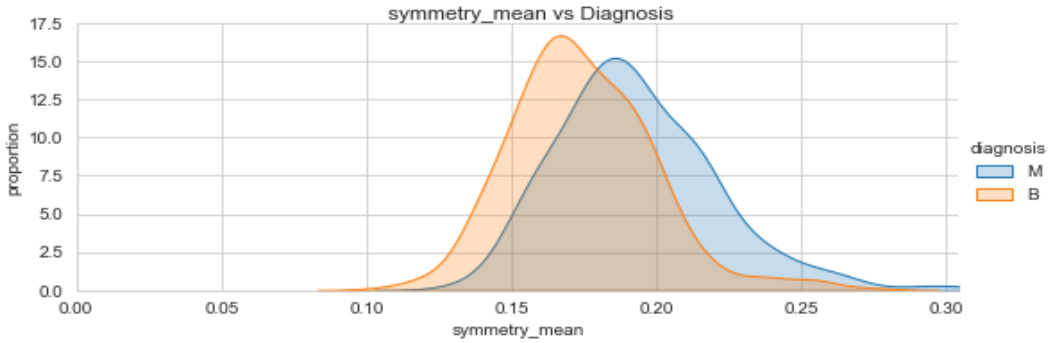
데이터 탐색 특징 별 분포

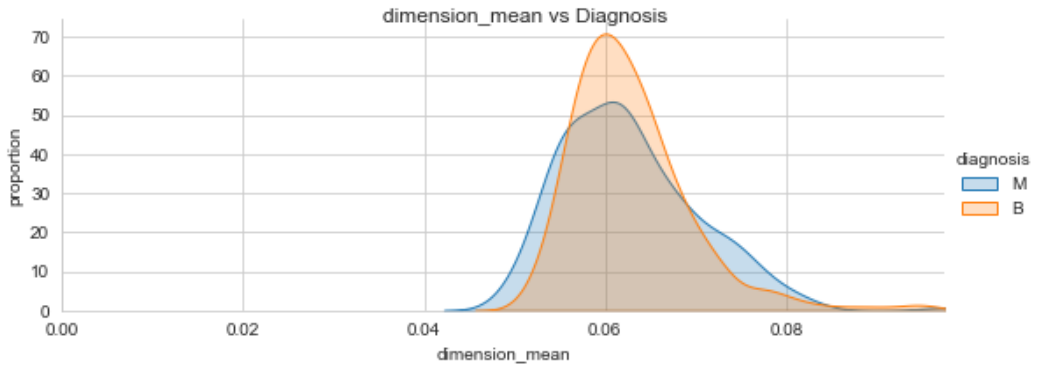




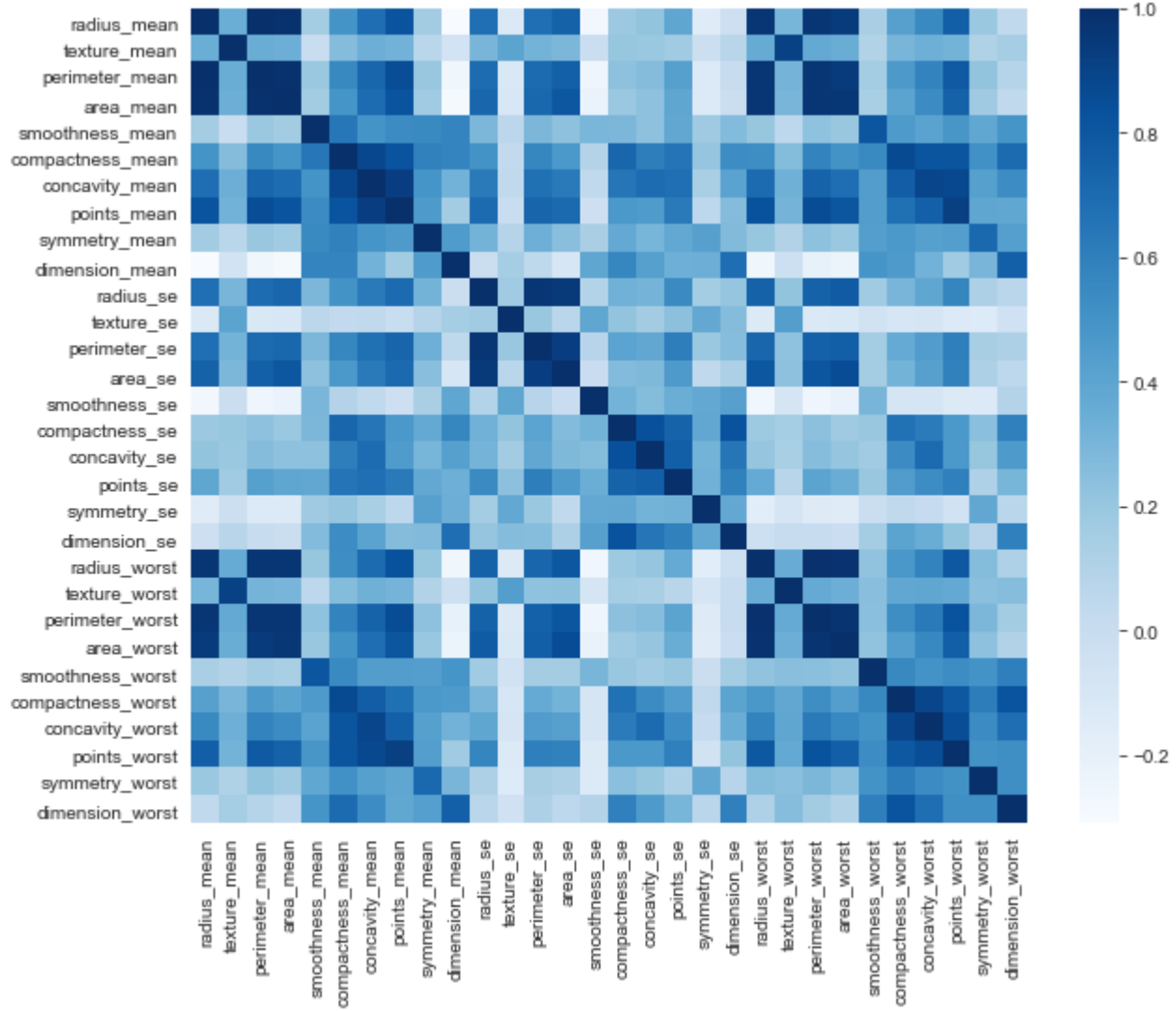








특징 선택 및 추출 전체 특징



특징이 많아지면 상관성을 갖는
특징을 수작업으로 찾기가 어렵다.

RFECV (Recursive Feature Elimination with Cross Validation)

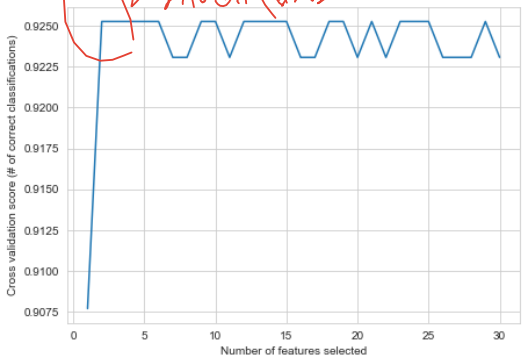
재귀적 특징 제거 방식

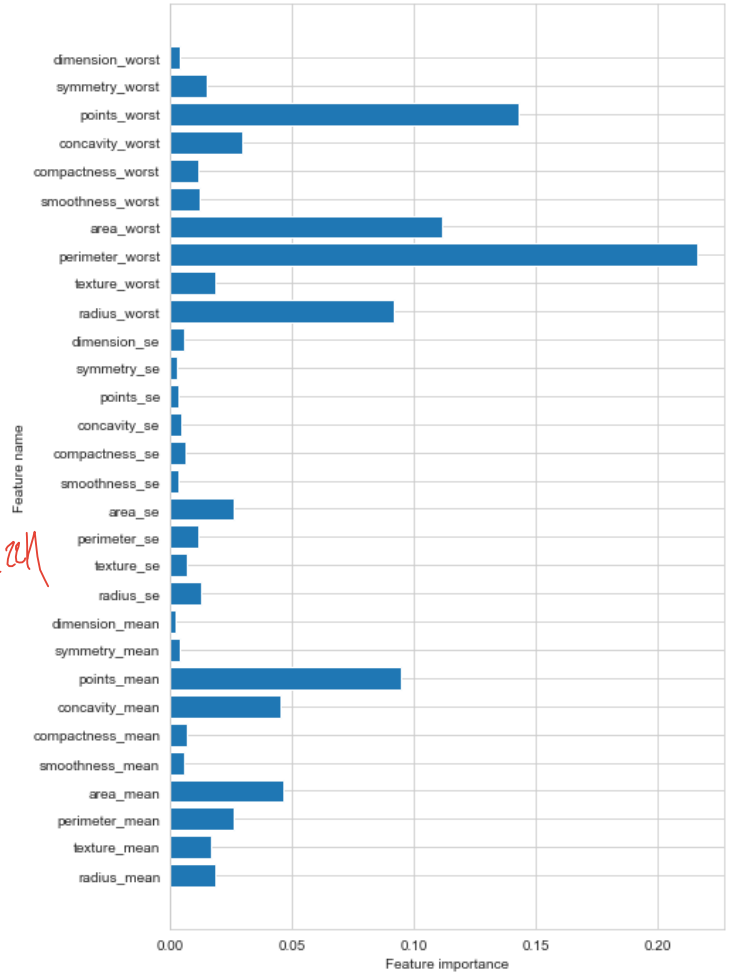
RFE (Recursive Feature Elimination)

- 학습을 할 때 특징의 중요도를 계산해서 중요도가 낮은 순서로 특징을 제거하면서 원하는 개수가 될 때까지 반복하는 방식
- 단, 특징을 몇 개 남겨야 할지 지정해야 함

RFECV (Recursive Feature Elimination with Cross Validation)

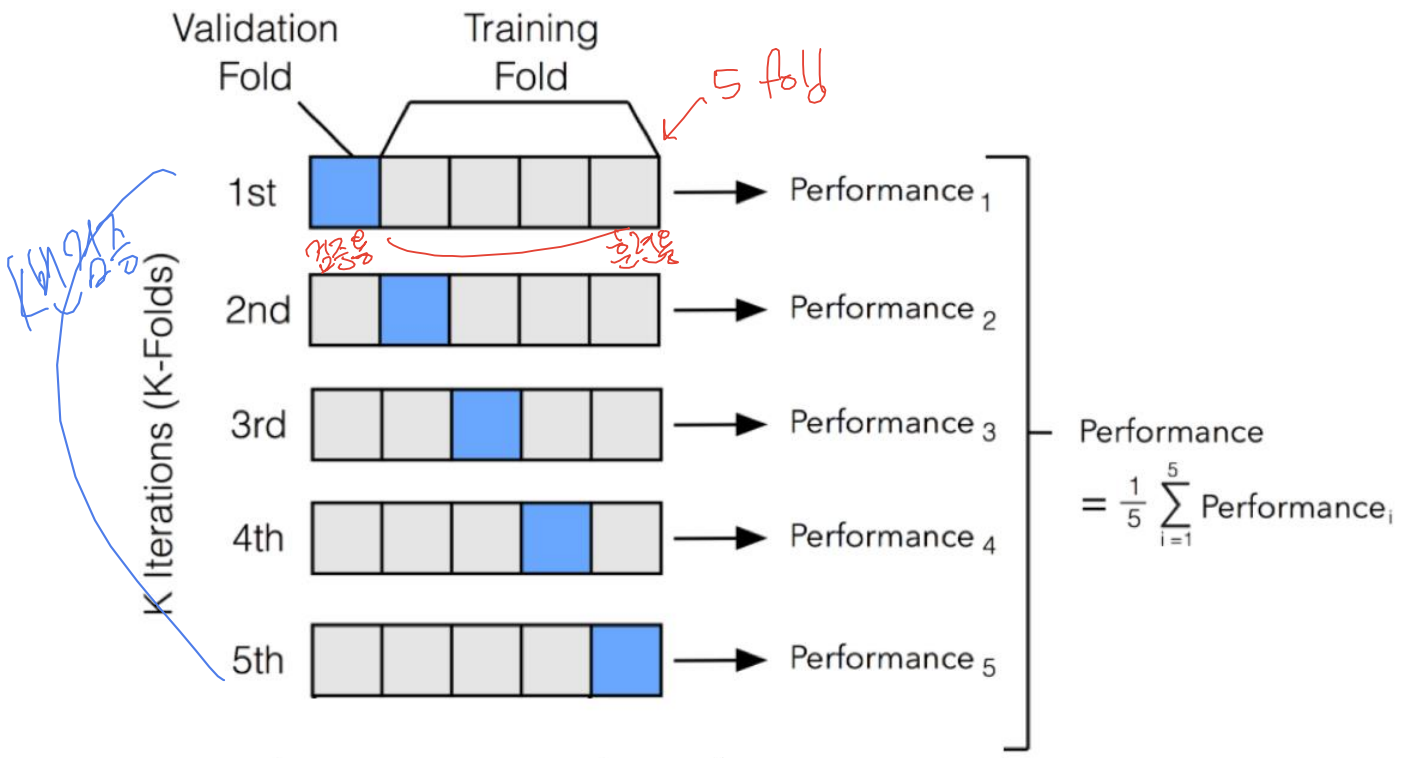
- RFE를 진행하면서 특징을 지울 때마다 모델의 성능을 교차 검증 방식으로 측정해서 남겨야 할 특징 개수를 정함





K-폴드 교차 검증 (K-fold Cross Validation)

데이터 분할에 따라 성능이 달라지므로 K번 훈련하고 검증해서 성능을 평균을 내는 방법



- 훈련 데이터를 K 등분
- K-1개 폴드는 훈련에 사용하고 나머지 1개 폴드는 검증에 사용
- 검증 폴드를 한 칸씩 이동하면서 전체 K번 훈련과 검증을 실시
- 훈련 및 검증 성능은 K번 실행한 성능을 평균으로 사용

차원 축소 특징 선택

RFECV 실행

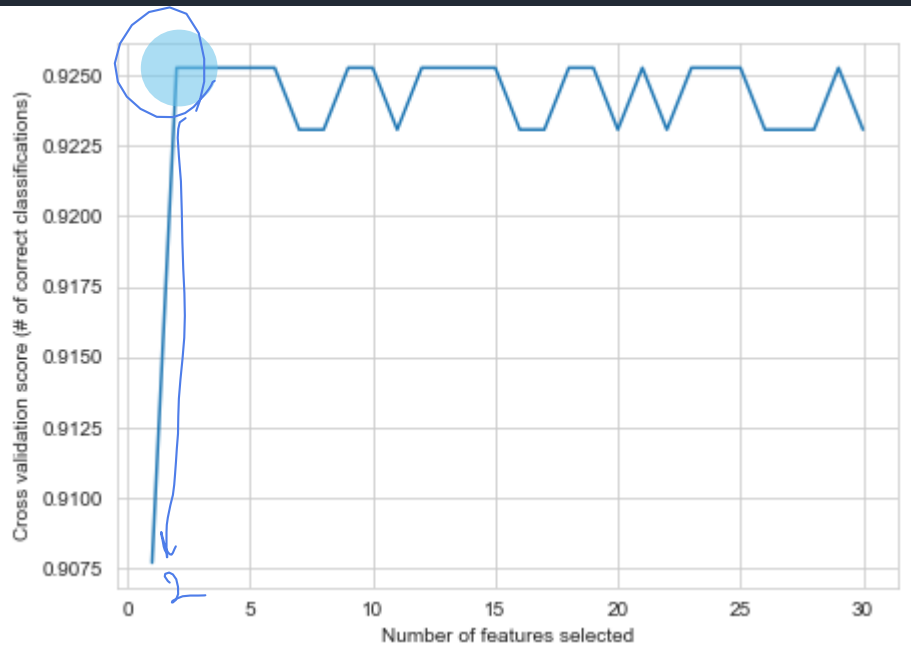
```
from sklearn.feature_selection import RFECV

min_features_to_select = 1
clf = DecisionTreeClassifier(max_depth=2, min_samples_leaf=12, random_state=12)
rfe = RFECV(estimator=clf,
            step=1,
            cv=5,                    # 5-fold cross-validation
            scoring='accuracy',
            min_features_to_select=min_features_to_select
           )
rfe = rfe.fit(X_train, Y_train)
```

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html

차원 축소 특징 선택

특징 선택을 위한 훈련 과정의 성능 그래프

```
# Plot number of features VS. cross-validation scores
plt.figure(figsize=(7,5))
plt.xlabel("Number of features selected")
plt.ylabel("Cross validation score (# of correct classifications)")
plt.plot(range(min_features_to_select, len(rfe.grid_scores_)+min_features_to_sel
ect),rfe.grid_scores_)
plt.show()
```



차원 축소 특징 선택

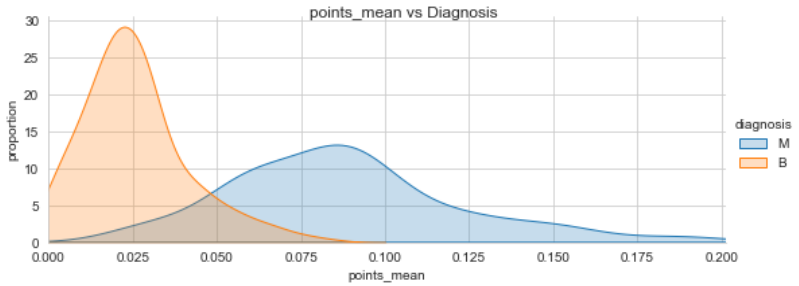
선택된 특징 확인

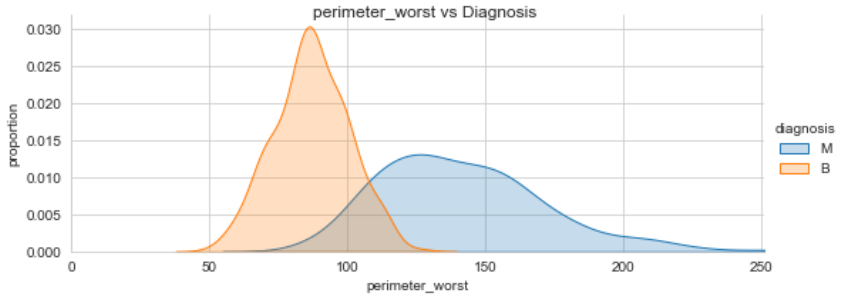
```
best_features = X_train.columns.values[rfe.support_]
drop_features = [ column_name for column_name in column_names[2:-
1] if column_name not in best_features ]

print('Optimal number of features :', rfe.n_features_)
print('Best features :', best_features)
print('Drop features :', drop_features)
```

Optimal number of features : **2**

Best features : **['points_mean' 'perimeter_worst']**

Drop features : ['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'symmetry_mean', 'dimension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'points_se', 'symmetry_se', 'dimension_se', 'radius_worst', 'texture_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'points_worst', 'symmetry_worst', 'dimension_worst']

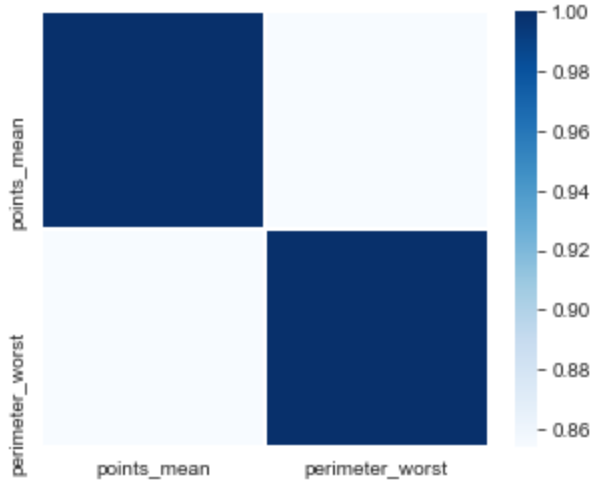




특징 선택 및 추출

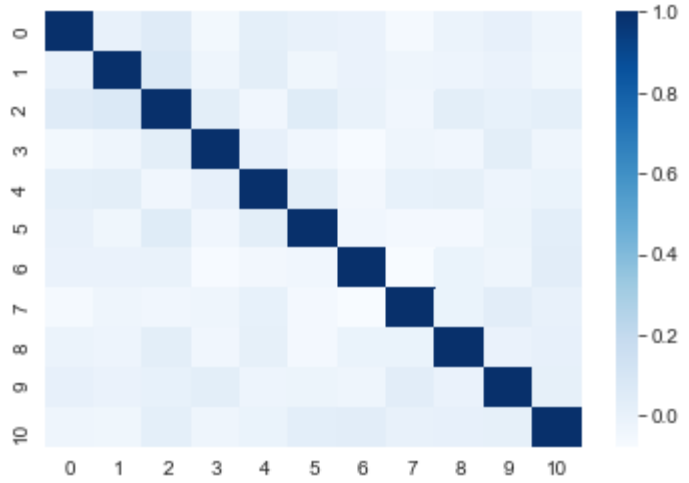
특징 선택

- 'points_mean', 'perimeter_worst'



특징 추출

- PCA로 11개의 특징을 추출

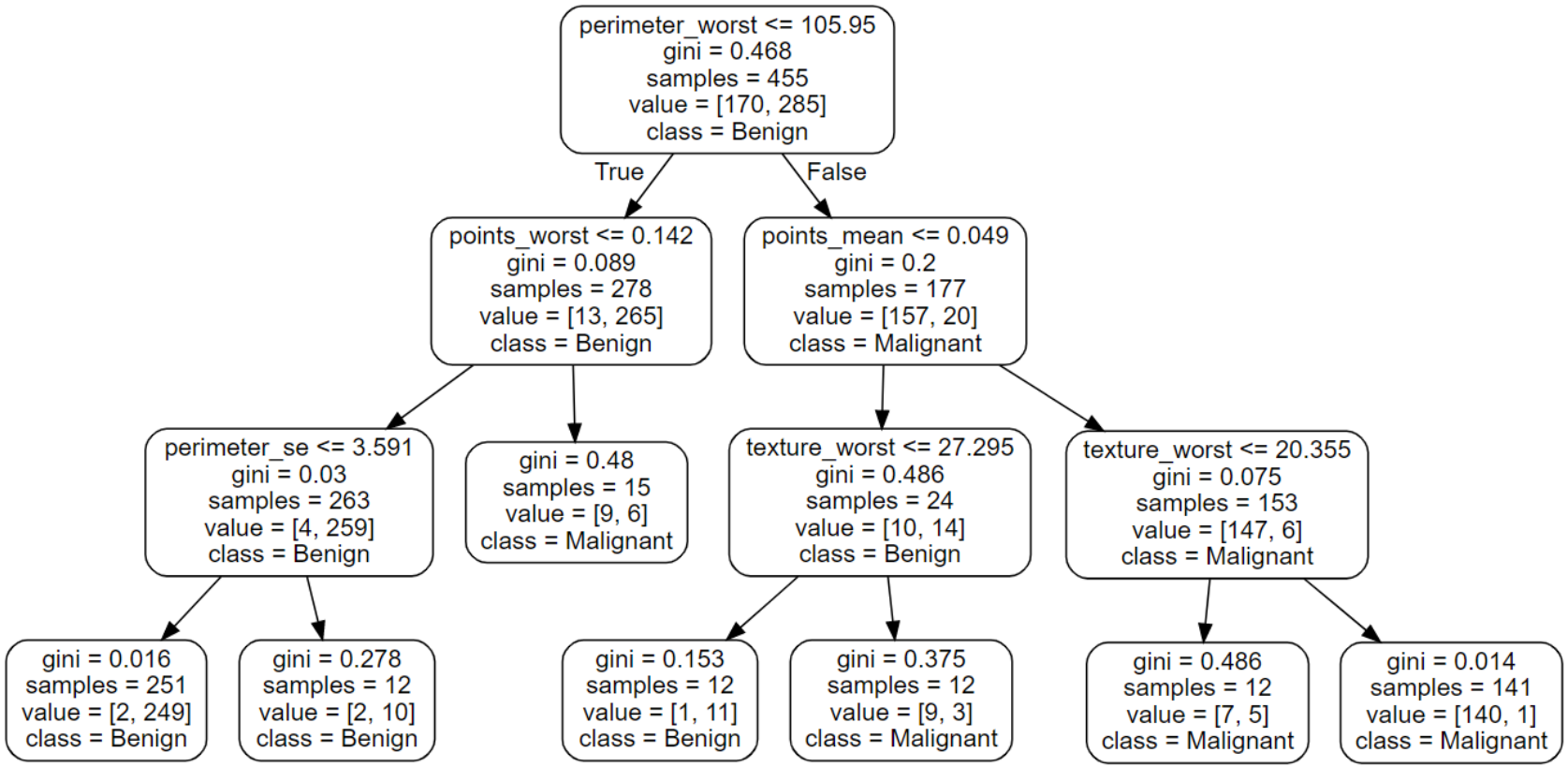


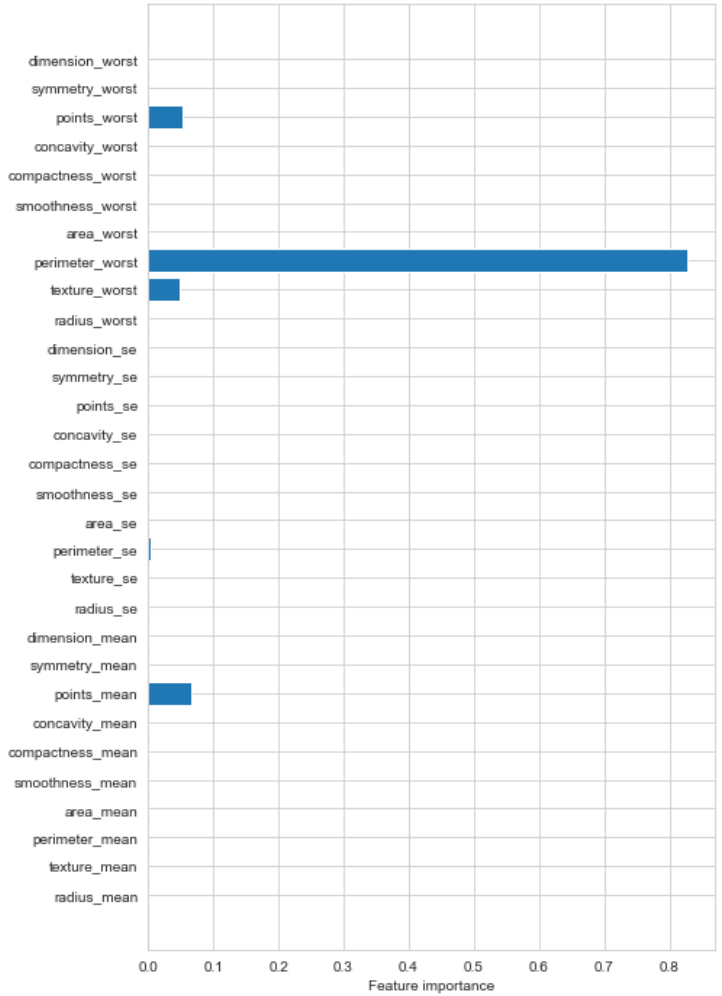
세 모델의 성능

최대 트리 깊이 = 3, 리프 노드 샘플 수 = 12

```
Accuracy of Decision Tree classifier on original training set: 0.96
Accuracy of Decision Tree classifier on original test set: 0.92
Accuracy of Decision Tree classifier on reduced training set: 0.95
Accuracy of Decision Tree classifier on reduced test set: 0.90
Accuracy of Decision Tree classifier on PCA-transformed training set: 0.94
Accuracy of Decision Tree classifier on PCA-transformed test set: 0.92
```

전체 특징

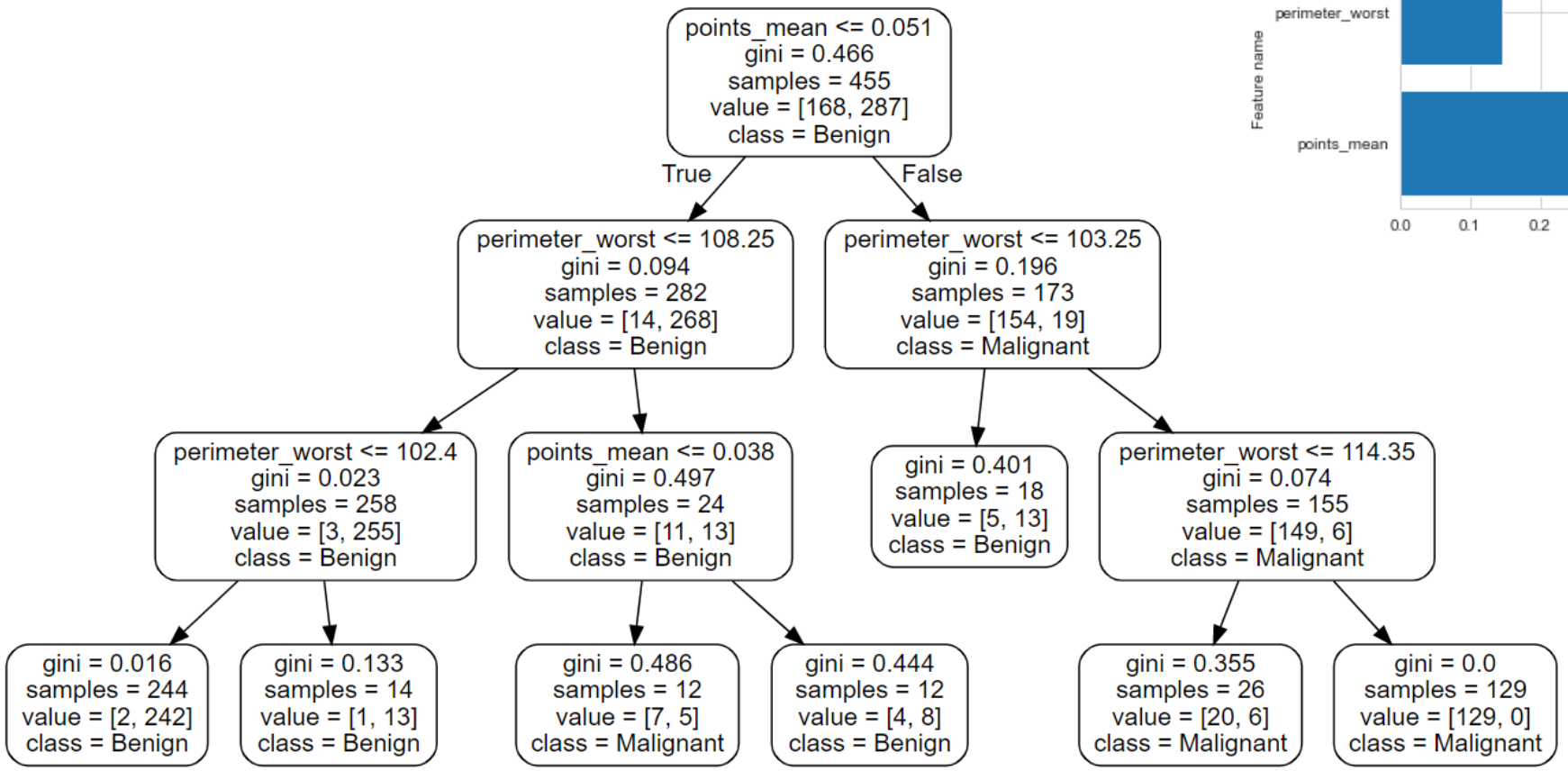


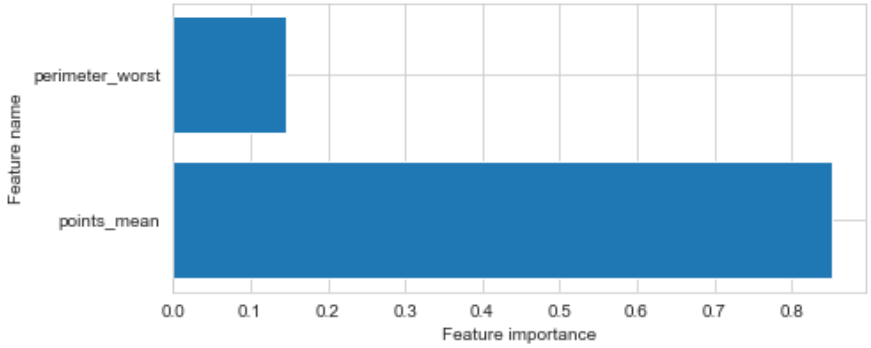



```
Accuracy of Decision Tree classifier on original training set: 0.96
Accuracy of Decision Tree classifier on original test set: 0.92
```

특징 선택 (RFECV)

트리 깊이를 3으로 지정했을 때

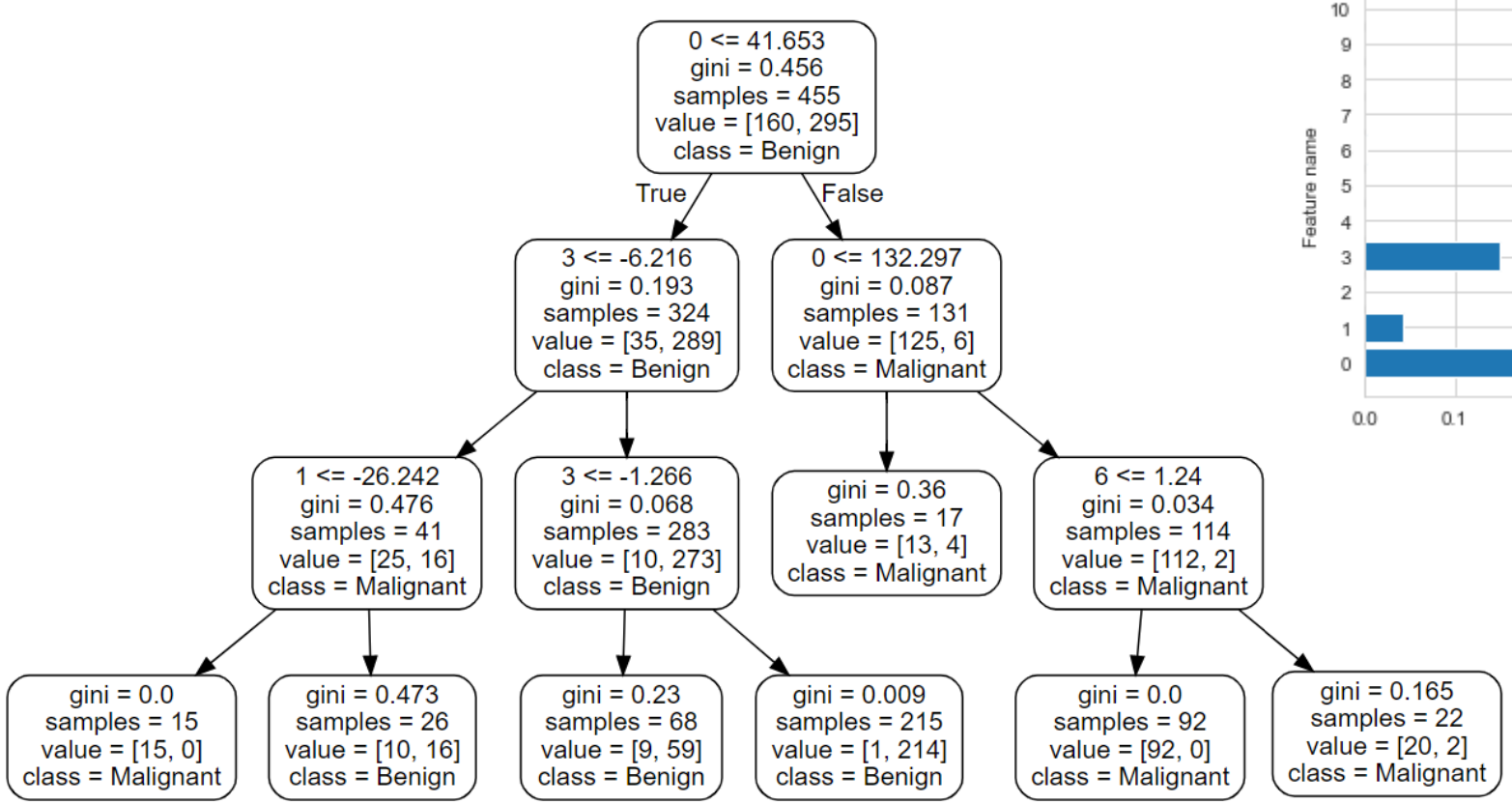


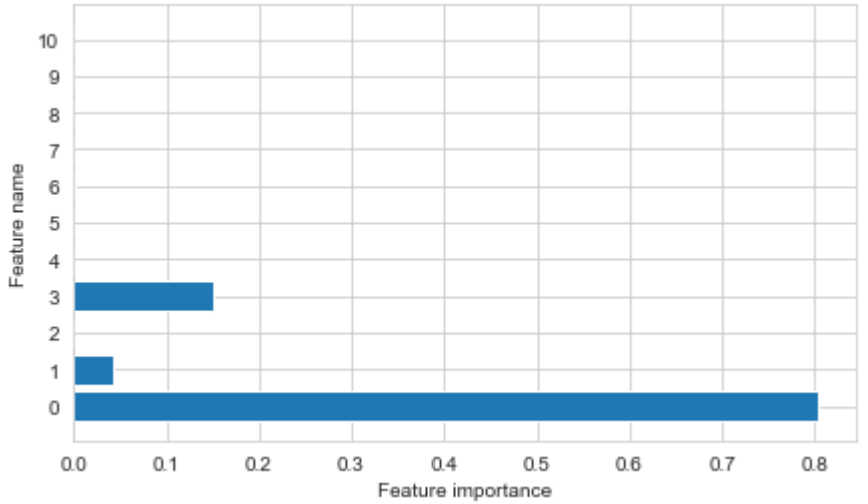



```
Accuracy of Decision Tree classifier on reduced training set: 0.95
Accuracy of Decision Tree classifier on reduced test set: 0.90
```

특징 추출 (PCA)

트리 깊이를 3으로 지정했을 때

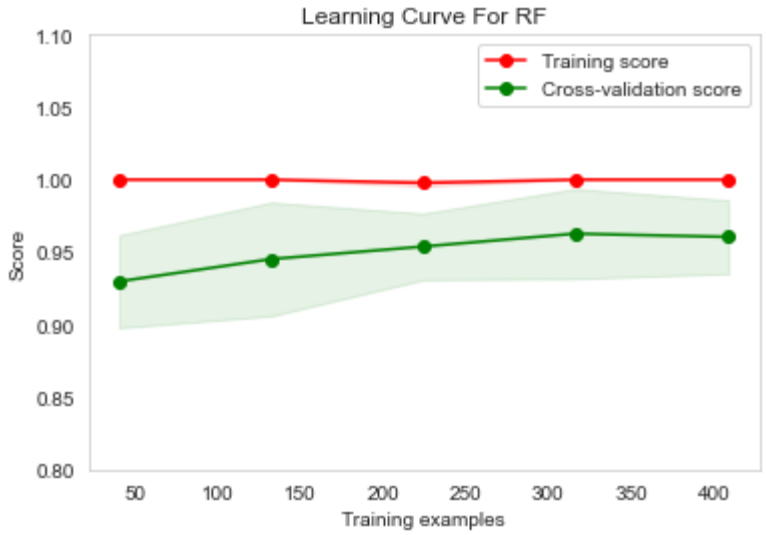


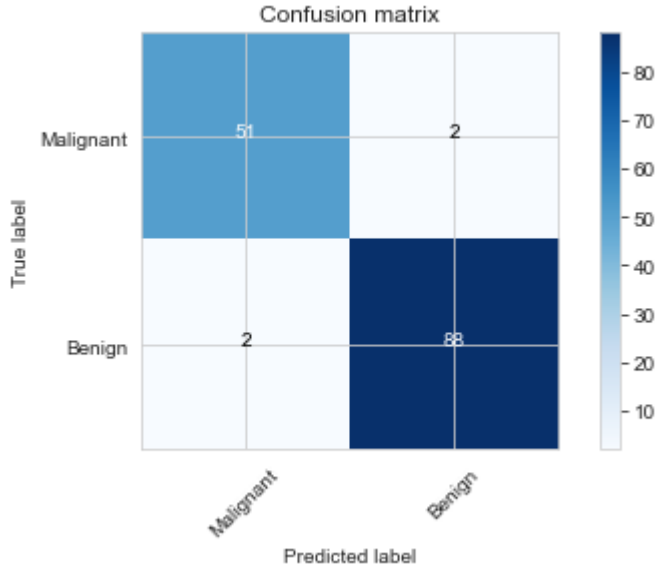



```
Accuracy of Decision Tree classifier on PCA-transformed training set: 0.94
Accuracy of Decision Tree classifier on PCA-transformed test set: 0.92
```

랜덤 포레스트

랜덤 포레스트를 적용하면 정확도가 97%까지 향상되는 것을 확인해보자.






```
Accuracy of Random Forest Classifier on training data: 1.00
Accuracy of Random Forest Classifier on testing data: 0.97
```

Thank you!