T. C.

ERCİYES ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

MACHINE LEARNING PROJE ÖDEVİ

HAZIRLAYAN:

1030520852 AYDAN ALKAYA

MAYIS 2022

Data:

We have blood sample data of 355 people with 4 most common cancer types: Colon cancer, breast cancer,lung cancer, and prostate cancer.You are given a label file, labels.csv, indicating the sample names, and the disease type of each person with the corresponding sample name. The data are stored in data.csv. Again, each row has the sample name ofthe corresponding person, and the remaining are the number of DNA fragments belonging to each microor-ganism type (virus or bacteria). 1836 different microorganisms appear as features.

Data link: https://drive.google.com/file/d/15evTOZTYuopoBnolYWOPy2P_VF6wnlFm/view

Classification Algorithms:
The classifications are going to be performed in Random Forest and Gradient Boosted Trees. The performance of these two algorithms are going to be compared. I used Python Programming Language.As the Gradient boosted tree algorithms, i used XGBoost.

Random Forest vs Gradient Boosting

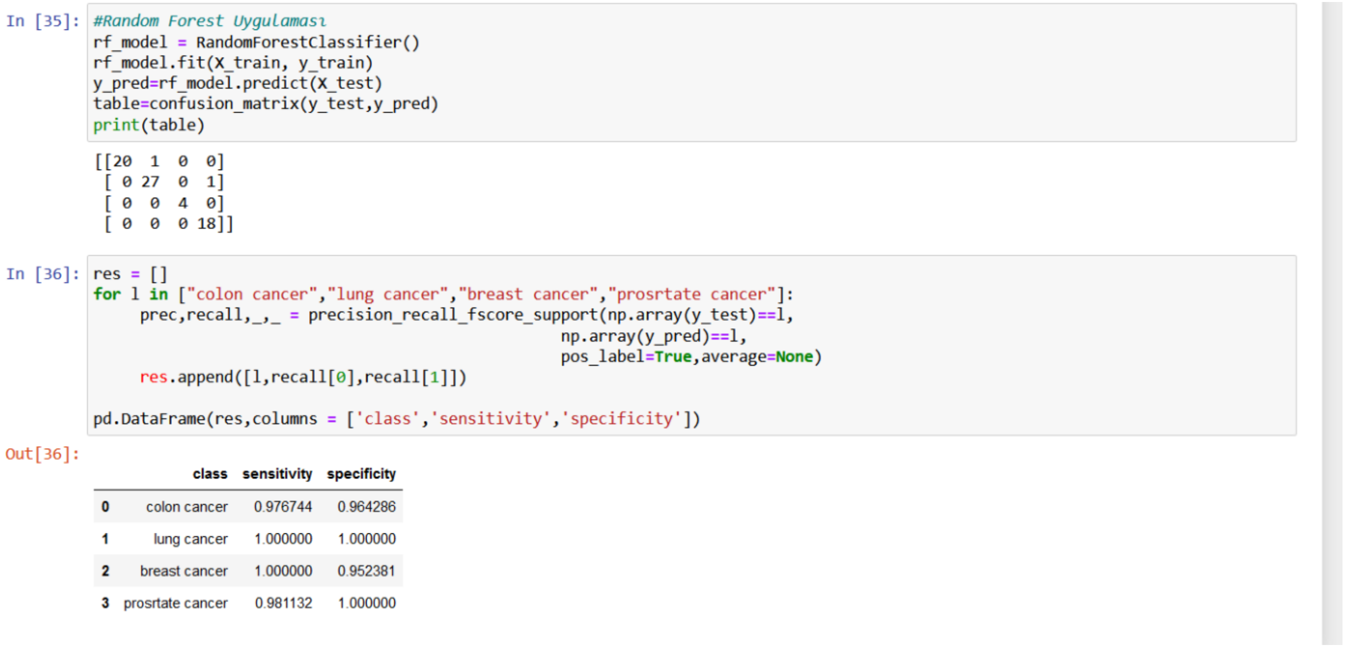
Like random forests, gradient boosting is a set of decision trees. The two main differences are:

1. **How trees are built:** random forests builds each tree independently while gradient boosting builds one tree at a time. This additive model (ensemble) works in a forward stage-wise manner, introducing a weak learner to improve the shortcomings of existing weak learners.
2. **Combining results**: random forests combine results at the end of the process (by averaging or "majority rules") while gradient boosting combines results along the way.

If you carefully tune parameters, gradient boosting can result in **better performance** than random forests. However, **gradient boosting may not be a good choice if you have a lot of noise**, as it can result in overfitting. They also tend to be **harder to tune** than random forests.

Random forests and gradient boosting each excel in different areas. Random forests perform well for multi-class object detection and bioinformatics, which tends to have a lot of statistical noise. Gradient Boosting performs well when you have unbalanced data such as in real time risk assessment.

Random Forest:



```
In [35]: #Random Forest Uygulaması
         rf_model = RandomForestClassifier()
         rf_model.fit(X_train, y_train)
         y_pred=rf_model.predict(X_test)
         table=confusion_matrix(y_test,y_pred)
         print(table)

         [[20  1  0  0]
          [ 0 27  0  1]
          [ 0  0  4  0]
          [ 0  0  0 18]]
```

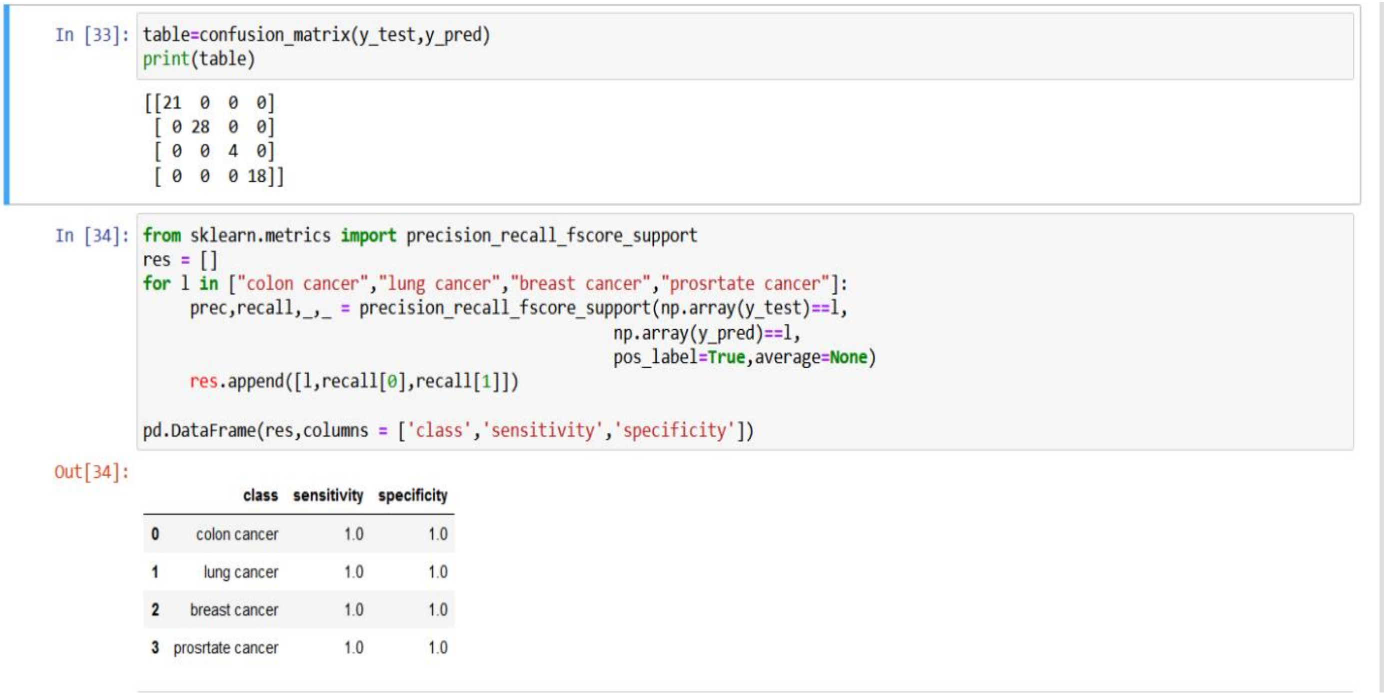
```
In [36]: res = []
         for l in ["colon cancer","lung cancer","breast cancer","prosrtate cancer"]:
             prec,recall,_,_ = precision_recall_fscore_support(np.array(y_test)==l,
                                                               np.array(y_pred)==l,
                                                               pos_label=True,average=None)
             res.append([l,recall[0],recall[1]])

         pd.DataFrame(res,columns = ['class','sensitivity','specificity'])
```

Out[36]:

	class	sensitivity	specificity
0	colon cancer	0.976744	0.964286
1	lung cancer	1.000000	1.000000
2	breast cancer	1.000000	0.952381
3	prosrtate cancer	0.981132	1.000000

Gradient Boosting:



```
In [33]: table=confusion_matrix(y_test,y_pred)
         print(table)

         [[21  0  0  0]
          [ 0 28  0  0]
          [ 0  0  4  0]
          [ 0  0  0 18]]
```

```
In [34]: from sklearn.metrics import precision_recall_fscore_support
         res = []
         for l in ["colon cancer","lung cancer","breast cancer","prosrtate cancer"]:
             prec,recall,_,_ = precision_recall_fscore_support(np.array(y_test)==l,
                                                               np.array(y_pred)==l,
                                                               pos_label=True,average=None)
             res.append([l,recall[0],recall[1]])

         pd.DataFrame(res,columns = ['class','sensitivity','specificity'])
```

Out[34]:

	class	sensitivity	specificity
0	colon cancer	1.0	1.0
1	lung cancer	1.0	1.0
2	breast cancer	1.0	1.0
3	prosrtate cancer	1.0	1.0

References:

https://drive.google.com/file/d/15evTOZTYuopoBnolYWOPy2P_VF6wnlFm/view

https://www.datasciencecentral.com/decision-tree-vs-random-forest-vs-boosted-trees-explained/

https://stackoverflow.com/questions/55635406/how-to-calculate-multiclass-overall-accuracy-sensitivity-and-specificity

https://www.veribilimiokulu.com/gradient-boosted-regresyon-agaclari/

https://xgboost.readthedocs.io/en/latest/parameter.html