raj_choudhary_16bce1384_boosting_adaboost_xgboost

September 18, 2018

0.1 Raj Choudhary

0.2 16BCE1384

0.3 Boosting –> Adaboost, XGboost

0.3.1 Importing the libraries

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline
```

0.3.2 Importing the dataset

```
In [2]: df_wine = pd.read_csv('wine.data',
                              header=None)

        df_wine.columns = ['Class label', 'Alcohol', 'Malic acid', 'Ash',
                           'Alcalinity of ash', 'Magnesium', 'Total phenols',
                           'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins',
                           'Color intensity', 'Hue', 'OD280/OD315 of diluted wines',
                           'Proline']

In [3]: df_wine = df_wine[df_wine['Class label'] != 1]

        y = df_wine['Class label'].values
        X = df_wine[['Alcohol', 'OD280/OD315 of diluted wines']].values
```

0.3.3 Preprocessing the dataset

```
In [4]: from sklearn.preprocessing import LabelEncoder
        from sklearn.model_selection import train_test_split

In [5]: le = LabelEncoder()
        y = le.fit_transform(y)

        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

0.3.4 Comparing the results of classification between Decision Tree Classifier and ADABoost Classifier

```
In [6]: from sklearn.ensemble import AdaBoostClassifier
        from sklearn.tree import DecisionTreeClassifier
```

```
In [7]: tree = DecisionTreeClassifier(criterion = 'entropy',
                                      max_depth = 1,
                                      random_state = 1)
        ada = AdaBoostClassifier(base_estimator = tree,
                                 n_estimators = 500,
                                 learning_rate = 0.1,
                                 random_state = 1)
```

```
In [8]: from sklearn.metrics import accuracy_score
```

```
In [9]: tree = tree.fit(X_train, y_train)
        y_train_pred = tree.predict(X_train)
        y_test_pred = tree.predict(X_test)

        tree_train = accuracy_score(y_train, y_train_pred)
        tree_test = accuracy_score(y_test, y_test_pred)
        print('Decision tree train/test accuracies %.3f/%.3f'
              % (tree_train, tree_test))
```

```
Decision tree train/test accuracies 0.916/0.875
```

```
In [10]: ada = ada.fit(X_train, y_train)
         y_train_pred = ada.predict(X_train)
         y_test_pred = ada.predict(X_test)

         ada_train = accuracy_score(y_train, y_train_pred)
         ada_test = accuracy_score(y_test, y_test_pred)
         print('AdaBoost train/test accuracies %.3f/%.3f'
               % (ada_train, ada_test))
```

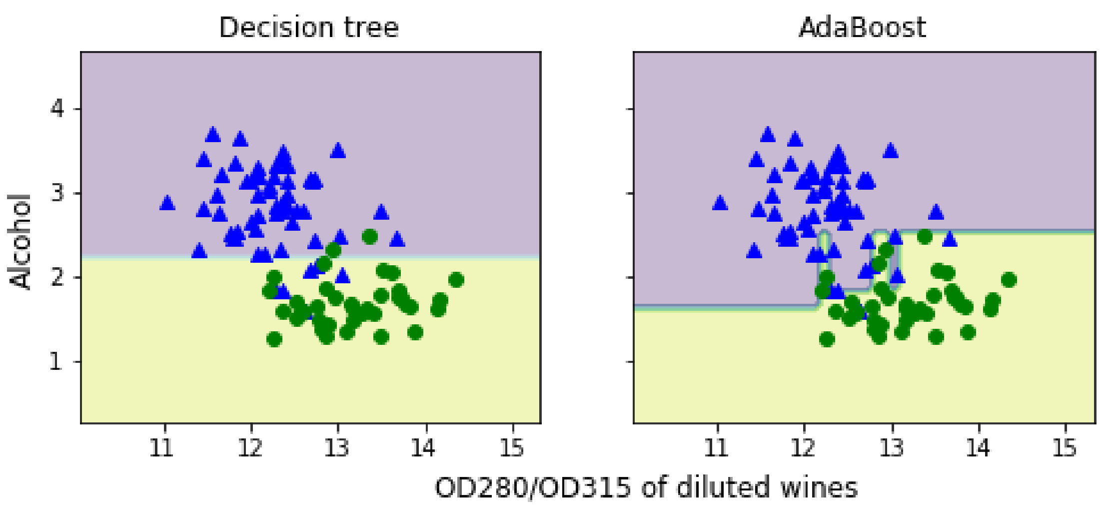
```
AdaBoost train/test accuracies 1.000/0.917
```

From the above train/test accuracies score, we can conclude that the AdaBoost Classfier is better able to generalize the pattern in the training set than Decision Tree Classifier as a result of which it it beter able to predict the results of the test set as can be seen from the test accuracy of 91.7% for AdaBoost Classifier as compared to 87.5% for Decision Tree Classifier

0.3.5 Plotting the decision boundary of Decision Tree Classifier and ADABoost Classifier

```
In [11]: x_min, x_max = X_train[:, 0].min() - 1, X_train[:, 0].max() + 1
         y_min, y_max = X_train[:, 1].min() - 1, X_train[:, 1].max() + 1
         xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.1),
```

```
                        np.arange(y_min, y_max, 0.1))

f, axarr = plt.subplots(1, 2, sharex='col', sharey='row', figsize=(8, 3))


for idx, clf, tt in zip([0, 1],
                        [tree, ada],
                        ['Decision tree', 'AdaBoost']):
    clf.fit(X_train, y_train)

    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)

    axarr[idx].contourf(xx, yy, Z, alpha=0.3)
    axarr[idx].scatter(X_train[y_train == 0, 0],
                       X_train[y_train == 0, 1],
                       c='blue', marker='^')
    axarr[idx].scatter(X_train[y_train == 1, 0],
                       X_train[y_train == 1, 1],
                       c='green', marker='o')
    axarr[idx].set_title(tt)

axarr[0].set_ylabel('Alcohol', fontsize=12)
plt.text(10.2, -0.5,
         s='OD280/OD315 of diluted wines',
         ha='center', va='center', fontsize=12)
plt.show()
```



0.3.6 Comparing the result of classification between Decision Tree Classifier and XGBoost Classifier

```
In [12]: from xgboost import XGBClassifier
```

```
In [13]: xgb = XGBClassifier(max_depth = 10,
                             learning_rate = 0.1,
                             n_estimators = 500,
                             n_jobs = -1)
```

```
In [14]: tree = tree.fit(X_train, y_train)
         y_train_pred = tree.predict(X_train)
         y_test_pred = tree.predict(X_test)

         tree_train = accuracy_score(y_train, y_train_pred)
         tree_test = accuracy_score(y_test, y_test_pred)
         print('Decision tree train/test accuracies %.3f/%.3f'
               % (tree_train, tree_test))
```

```
Decision tree train/test accuracies 0.916/0.875
```

```
In [15]: xgb = xgb.fit(X_train, y_train)
         y_train_pred = xgb.predict(X_train)
         y_test_pred = xgb.predict(X_test)

         xgb_train = accuracy_score(y_train, y_train_pred)
         xgb_test = accuracy_score(y_test, y_test_pred)
         print('XGBoost tree train/test accuracies %.3f/%.3f'
               % (xgb_train, xgb_test))
```

```
XGBoost tree train/test accuracies 0.989/0.917
```

```
/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning:
  if diff:
/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning:
  if diff:
```

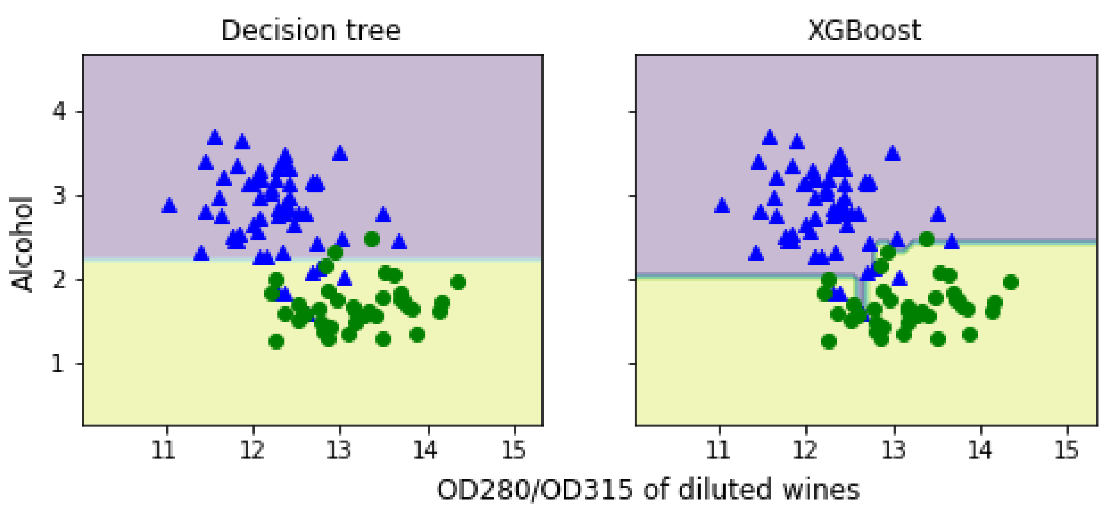
From the above train/test accuracies, we can conclude that as compared to Decision Tree Classifer, XGBoost Classifer is better able to recognize the pattern in the dataset, as a result of which, we see a better result in test set prediction accuracy i.e. 91.7% for the XGBoost Classifier whereas 87.5% for Decision Tree Classifier

0.3.7 Plotting the decision boundary of Decision Tree Classifier and XGBoost Classifier

```
In [16]: x_min, x_max = X_train[:, 0].min() - 1, X_train[:, 0].max() + 1
         y_min, y_max = X_train[:, 1].min() - 1, X_train[:, 1].max() + 1
         xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.1),
```

```
                         np.arange(y_min, y_max, 0.1))

f, axarr = plt.subplots(1, 2, sharex='col', sharey='row', figsize=(8, 3))


for idx, clf, tt in zip([0, 1],
                        [tree, xgb],
                        ['Decision tree', 'XGBoost']):
    clf.fit(X_train, y_train)

    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)

    axarr[idx].contourf(xx, yy, Z, alpha=0.3)
    axarr[idx].scatter(X_train[y_train == 0, 0],
                       X_train[y_train == 0, 1],
                       c='blue', marker='^')
    axarr[idx].scatter(X_train[y_train == 1, 0],
                       X_train[y_train == 1, 1],
                       c='green', marker='o')
    axarr[idx].set_title(tt)

axarr[0].set_ylabel('Alcohol', fontsize=12)
plt.text(10.2, -0.5,
         s='OD280/OD315 of diluted wines',
         ha='center', va='center', fontsize=12)
plt.show()
```

```
/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning:
  if diff:
```



From the above results, we can conclude that AdaBoost Classifier and XGBoost Classifier are able to perform better as compared to simple Decision Tree Classifier. We can also improve the performance of the two algorithms by tuning the respective hyperparameters via GridSearchCV