**RANDOM FOREST**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('Social\_Network\_Ads.csv')

df

x = df.iloc[:, :-1].values

y = df.iloc[:, -1].values

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.2, random\_state=0)

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

x\_train = sc.fit\_transform(x\_train)

x\_test = sc.transform(x\_test)

Training the RandomForest model on the training set

from sklearn.ensemble import RandomForestClassifier

classifier\_rf = RandomForestClassifier(n\_estimators=50, criterion='entropy', random\_state=0)

classifier\_rf.fit(x\_train, y\_train)

print(classifier\_rf.predict(sc.transform([[30,87000]])))

y\_pred = classifier\_rf.predict(x\_test)

print(np.concatenate((y\_test.reshape(len(y\_test),1), y\_pred.reshape(len(y\_pred),1)),1))

from sklearn.metrics import confusion\_matrix, accuracy\_score, f1\_score, recall\_score, precision\_score

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

accuracy\_score(y\_test, y\_pred)

visualizing training set results

from matplotlib.colors import ListedColormap

x\_set, y\_set = sc.inverse\_transform(x\_train), y\_train

x1, x2 = np.meshgrid(np.arange(start = x\_set[:, 0].min() - 10, stop = x\_set[:, 0].max() + 10, step = 1),

np.arange(start = x\_set[:, 1].min() - 1000, stop = x\_set[:, 1].max() + 1000, step = 1))

plt.contourf(x1, x2, classifier\_rf.predict(sc.transform(np.array([x1.ravel(), x2.ravel()]).T)).reshape(x1.shape),

alpha = 0.75, cmap = ListedColormap(('red', 'green')))

plt.xlim(x1.min(), x1.max())

plt.ylim(x2.min(), x2.max())

for i, j in enumerate(np.unique(y\_set)):

plt.scatter(x\_set[y\_set == j, 0], x\_set[y\_set == j, 1], c = ListedColormap(('red', 'green'))(i), label = j)

plt.title('Random Forest (Training set)')

plt.xlabel('Age')

plt.ylabel('Estimated Salary')

plt.legend()

plt.show()

visualizing test set results

from matplotlib.colors import ListedColormap

x\_set, y\_set = sc.inverse\_transform(x\_test), y\_test

x1, x2 = np.meshgrid(np.arange(start = x\_set[:, 0].min() - 10, stop = x\_set[:, 0].max() + 10, step = 1),

np.arange(start = x\_set[:, 1].min() - 1000, stop = x\_set[:, 1].max() + 1000, step = 1))

plt.contourf(x1, x2, classifier\_rf.predict(sc.transform(np.array([x1.ravel(), x2.ravel()]).T)).reshape(x1.shape),

alpha = 0.75, cmap = ListedColormap(('red', 'green')))

plt.xlim(x1.min(), x1.max())

plt.ylim(x2.min(), x2.max())

for i, j in enumerate(np.unique(y\_set)):

plt.scatter(x\_set[y\_set == j, 0], x\_set[y\_set == j, 1], c = ListedColormap(('red', 'green'))(i), label = j)

plt.title('Random Forest (Test set)')

plt.xlabel('Age')

plt.ylabel('Estimated Salary')

plt.legend()

plt.show()

**ADABOOST**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.ensemble import AdaBoostClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

df = pd.read\_csv('Social\_Network\_Ads.csv')

df

df['EstimatedSalary'] = df['EstimatedSalary']/1000

df

x = df.iloc[:, :-1].values

y = df.iloc[:, -1].values

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.2, random\_state=0)

#create AdaBoost classifier model object

AdaModel = AdaBoostClassifier(n\_estimators=100)

AdaModel.fit(x\_train, y\_train)

y\_pred = AdaModel.predict(x\_test)

print(np.concatenate((y\_test.reshape(len(y\_test),1), y\_pred.reshape(len(y\_pred),1)),1))

from sklearn.metrics import confusion\_matrix, accuracy\_score, f1\_score, recall\_score, precision\_score

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

accuracy\_score(y\_test, y\_pred)

f1\_score(y\_test, y\_pred)

visualizing the training set results

from matplotlib.colors import ListedColormap

x\_set, y\_set = x\_train, y\_train

x1, x2 = np.meshgrid(np.arange(start = x\_set[:, 0].min() - 10, stop = x\_set[:, 0].max() + 10, step = 1),

np.arange(start = x\_set[:, 1].min() - 1000, stop = x\_set[:, 1].max() + 1000, step = 1))

plt.contourf(x1, x2, AdaModel.predict((np.array([x1.ravel(), x2.ravel()]).T)).reshape(x1.shape),

alpha = 0.75, cmap = ListedColormap(('red', 'green')))

plt.xlim(x1.min(), x1.max())

plt.ylim(x2.min(), x2.max())

for i, j in enumerate(np.unique(y\_set)):

plt.scatter(x\_set[y\_set == j, 0], x\_set[y\_set == j, 1], c = ListedColormap(('red', 'green'))(i), label = j)

plt.title('AdaBoost (Training set)')

plt.xlabel('Age')

plt.ylabel('Estimated Salary')

plt.legend()

plt.show()

visualizing the test set results

from matplotlib.colors import ListedColormap

x\_set, y\_set = x\_test, y\_test

x1, x2 = np.meshgrid(np.arange(start = x\_set[:, 0].min() - 10, stop = x\_set[:, 0].max() + 10, step = 1),

np.arange(start = x\_set[:, 1].min() - 1000, stop = x\_set[:, 1].max() + 1000, step = 1))

plt.contourf(x1, x2, AdaModel.predict((np.array([x1.ravel(), x2.ravel()]).T)).reshape(x1.shape),

alpha = 0.75, cmap = ListedColormap(('red', 'green')))

plt.xlim(x1.min(), x1.max())

plt.ylim(x2.min(), x2.max())

for i, j in enumerate(np.unique(y\_set)):

plt.scatter(x\_set[y\_set == j, 0], x\_set[y\_set == j, 1], c = ListedColormap(('red', 'green'))(i), label = j)

plt.title('AdaBoost (Test set)')

plt.xlabel('Age')

plt.ylabel('Estimated Salary')

plt.legend()

plt.show()