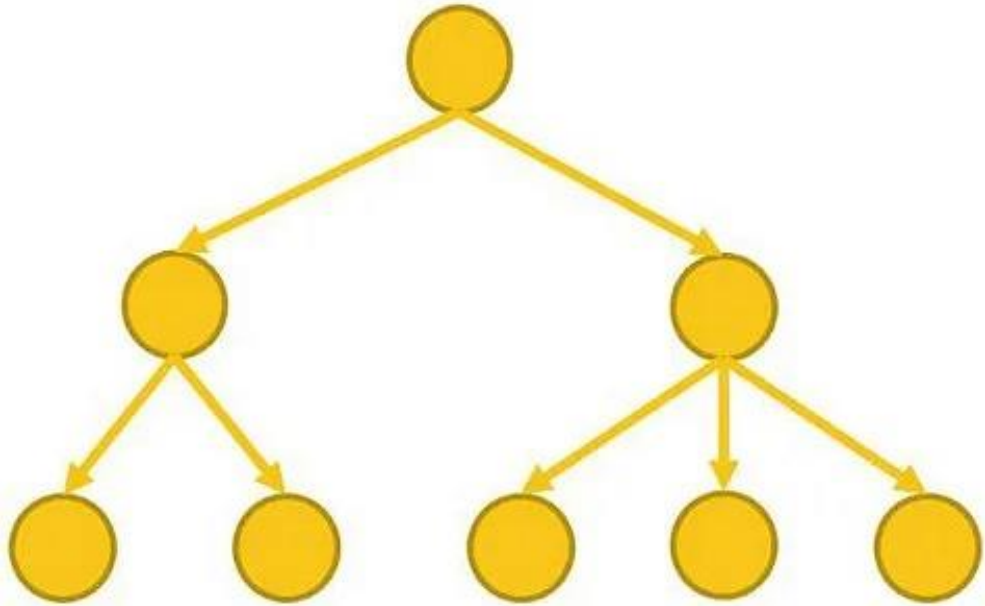


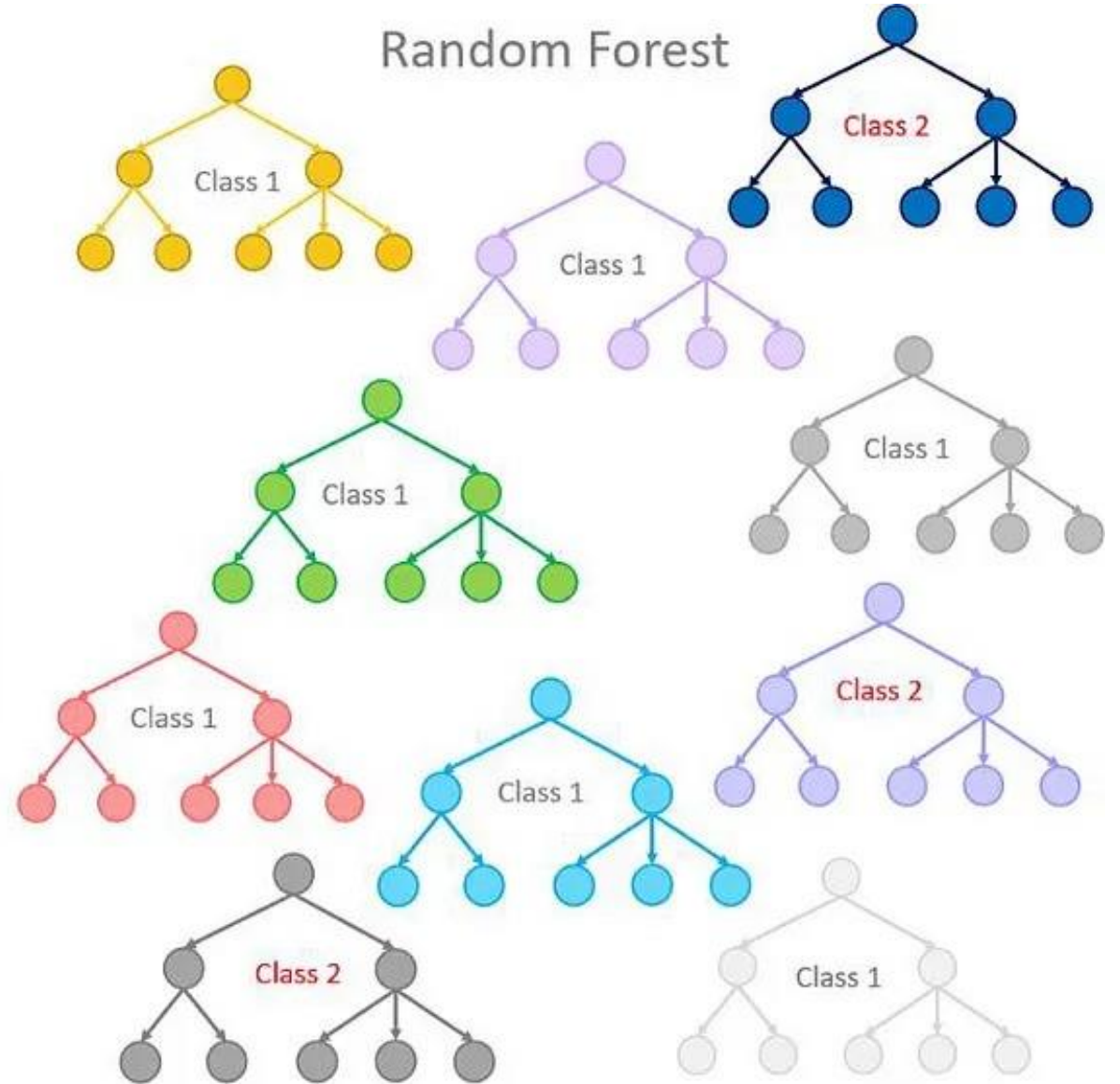
Random Forest (Rassal Orman) Algoritması

- Bu yazımızda makine öğrenmesi algoritmalarından biri olan random forest(rassal orman) algoritması üzerine bilgi aktarımında bulunacağız.
- Konuya kavramsal bir açıklamayla başlayalım.
- Random Forest (Rassal Orman) Algoritması:
- Sınıflandırma işlemi esnasında birden fazla karar ağacı üreterek sınıflandırma değerini yükseltmeyi hedefleyen bir algoritmadır. Bireysel olarak oluşturulan karar ağaçları bir araya gelerek karar ormanı oluşturur. Buradaki karar ağaçları bağlı olduğu veri setinden rastgele seçilmiş birer alt kümedir

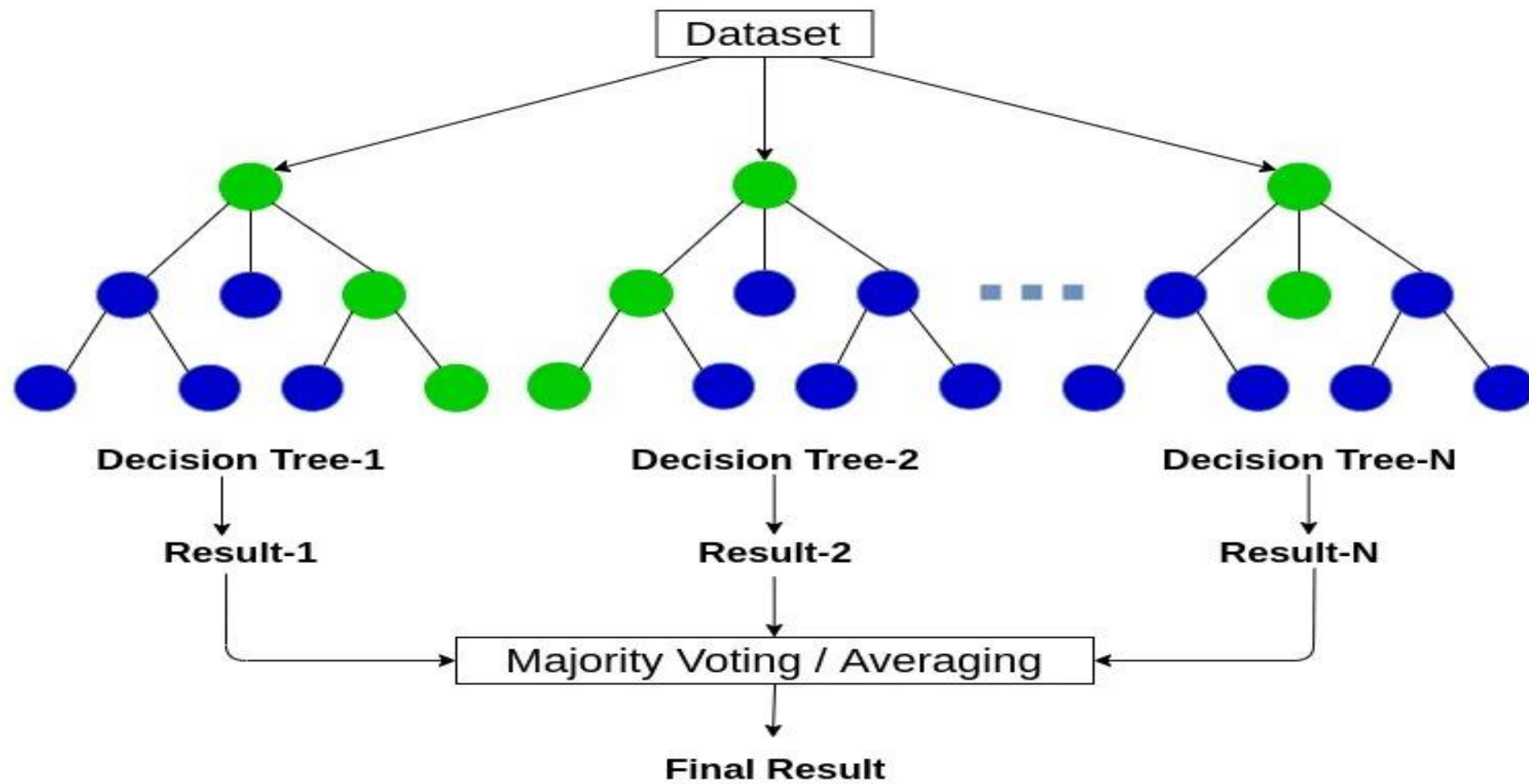
Single Decision Tree



Random Forest



- Random Forest Algoritmasının özelliklerini listelersek
 - Tek bir ağacın aksine birden fazla ağaç yetiştirilir.
 - Sınıflandırma durumunda ormandaki tüm ağaçlar arasında en çok oyu alan seçilir.
 - Regresyon durumunda, farklı ağaçların çıktılarının ortalamasını alır.
 - Bu özellikler algoritmanın kararlılığını artırır ve mevcut makine öğrenmesi sistemlerinin çoğunu oluşturan sınıflandırma ve regresyon problemlerinin çözümü için kullanılabilir. Ancak regresyon problemlerinin çözümünde sınıflandırma problemlerindeki kadar iyi değildir. Başka bir dezavantajı da hızlı eğitilebilir olmasının yanı sıra bir kez eğitildiklerinde tahminler oluşturmak için oldukça yavaştır.
- Random Forest Algoritması,
 - Görüntü sınıflandırması
 - Nesne algılama
 - Nesne takibi
 - Kenar algılama
 - Anlamsal bölümleme gibi yerlerde çokça kullanılır.



Örnek

- # Random Forest
- # Kütüphaneler içe aktarılır
- import numpy as np
- import matplotlib.pyplot as plt
- import pandas as pd
- # Veri seti içe aktarılır, bağımlı ve bağımsız değişkenler belirlenir
- datasets = pd.read_csv('/home/deniz/Masaüstü/random forest/Social_Network_Ads.csv')
- X = datasets.iloc[:, [2,3]].values
- Y = datasets.iloc[:, 4].values

- # Veri setini eğitim ve test seti olarak ayırma
- from sklearn.model_selection import train_test_split
- X_Train, X_Test, Y_Train, Y_Test = train_test_split(X, Y, test_size = 0.25, random_state = 0)
- # Nitelikleri Ölçeklendirme
- from sklearn.preprocessing import StandardScaler
- sc_X = StandardScaler()
- X_Train = sc_X.fit_transform(X_Train)
- X_Test = sc_X.transform(X_Test)

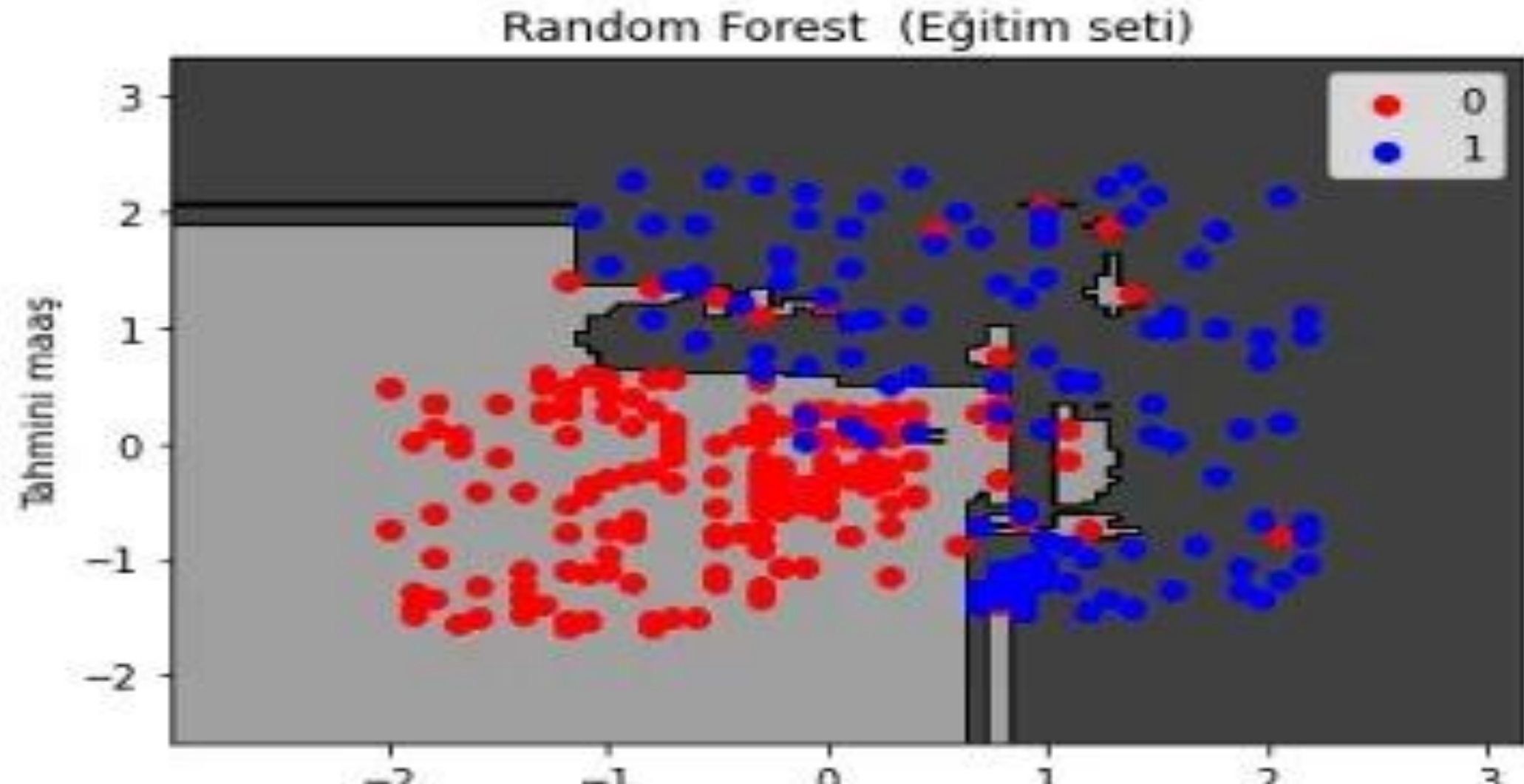
- # Eğitim setine karar ağacı uygulanması
- from sklearn.ensemble import RandomForestClassifier
- classifier = RandomForestClassifier(n_estimators = 200, criterion = 'entropy', random_state = 0)
- classifier.fit(X_Train,Y_Train)
- # Denemeyi tahmin etme aşaması
- Y_Pred = classifier.predict(X_Test)
- # Confusion Matrisi oluşturulur
- from sklearn.metrics import confusion_matrix
- cm = confusion_matrix(Y_Test, Y_Pred)

- # Oluşturulan eğitim seti görsellenir
- from matplotlib.colors import ListedColormap
- X_Set, Y_Set = X_Train, Y_Train
- X1, X2 = np.meshgrid(np.arange(start = X_Set[:, 0].min() - 1, stop = X_Set[:, 0].max() + 1, step = 0.01),
- np.arange(start = X_Set[:, 1].min() - 1, stop = X_Set[:, 1].max() + 1, step = 0.01))
- plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
- alpha = 0.75, cmap =
- ListedColormap(('grey', 'black')))
- 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', 'blue'))(i), label = j)
- plt.title('Random Forest (Eğitim seti)')
- plt.xlabel('Yaş')
- plt.ylabel('Tahmini maaş')
- plt.legend()
- plt.show()

•# Oluşturulan test seti görsellenir

- from matplotlib.colors import ListedColormap
- X_Set, Y_Set = X_Test, Y_Test
- X1, X2 = np.meshgrid(np.arange(start = X_Set[:, 0].min() - 1, stop = X_Set[:, 0].max() + 1, step = 0.01),
- np.arange(start = X_Set[:, 1].min() - 1, stop = X_Set[:, 1].max() + 1, step = 0.01))
- plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
- alpha = 0.75, cmap = ListedColormap(('grey', 'black')))
- 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', 'blue'))(i), label = j)
- plt.title('Random Forest (Deneme seti)')
- plt.xlabel('Yaş')
- plt.ylabel('Tahmini maaş')
- plt.legend()
- plt.show()

Çıktı



Random Forest (Deneme seti)

