



Bursa Teknik Üniversitesi
Mühendislik ve doğa bilimleri Fakültesi
Bilgisayar Mühendisi
Veri Madenciliği
(BLM0463)

Proje Adı: Extra Virgin Olive Oil (Sızma zeytinyağı)
Veri Seti Decision Tree ile İncelenmesi

FARHAN AHMAD
20360859096



Özet:

Bu proje, süzme ve organik zeytinyağlarının analizini ve sınıflandırmasını amaçlamaktadır. Veri toplama aşamasında, farklı zeytinyağı türlerinin kimyasal ve fiziksel özellikleri incelenmiştir. Toplanan veriler, veri temizleme ve ön işleme aşamalarından geçirilmiştir. Daha sonra, zeytinyağlarını sınıflandırmak için **Decision Tree Classifier** ve **Gaussian Mixture Model** kullanılmıştır. Modellerin performansları değerlendirilmiş ve sonuçlar, süzme ve organik zeytinyağlarının belirgin farklılıklarını doğru bir şekilde tespit edebilmiştir. Bu çalışma, zeytinyağı üreticilerine ve tüketicilerine ürünlerin kalitesini ve türünü belirlemede faydalı bilgiler sağlamayı hedeflemektedir.

Amaç:

Bu projenin temel amacı, süzme ve organik süzme zeytinyağlarının kimyasal bileşenlerine dayanarak sınıflandırılmasını sağlamaktır. Projede, zeytinyağı türlerinin doğru bir şekilde tanımlanması ve sınıflandırılması hedeflenmektedir. Bu sınıflandırma işlemi, zeytinyağlarının kalitesini belirlemede ve pazarlama stratejilerinde önemli rol oynayacaktır.

Kullanılan Yöntemler:

- **Decision Tree Classifier:** Karar ağaçları, veriyi dallara ayırarak sınıflandırma işlemi yapar. Bu projede, karar ağaçları kullanılarak zeytinyağı türleri arasındaki farklar belirlenmiştir.
- **Gaussian Mixture Model (GMM):** GMM, veriyi normal dağılımlar ile modelleyerek sınıflandırma yapar. Zeytinyağlarının kimyasal bileşenleri üzerinden farklı türlerin dağılımı incelenmiştir.

Metodoloji:

Veri Seti Bilgileri:

Başlangıçta veri seti şu sütunlardan oluşmaktadır:

- ID
- Bölge
- Alan
- Palmitik (*)
- Palmitoleik (*)
- Stearik (*)
- Oleik (*)
- Linoleik (*)
- Linolenik (*)
- Arachidik (*)
- Eikosenoik (*)
- Diğer

Toplamda 572 örnek bulunmaktadır.

Yağ Asitleri:

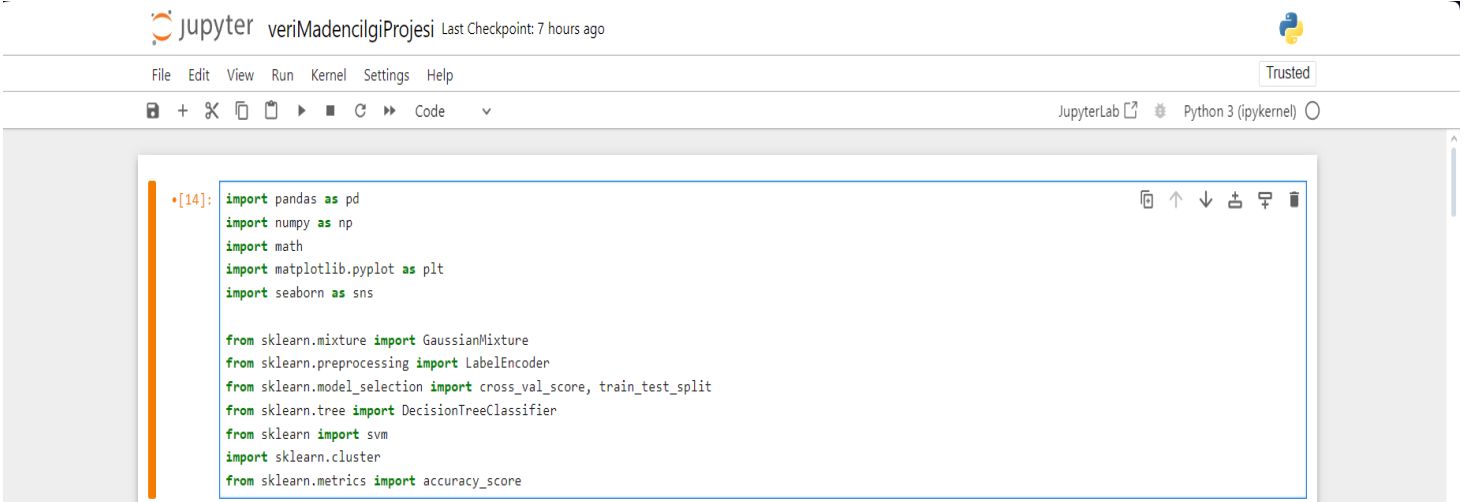
Veri setindeki (*) ile işaretlenmiş alanlar zeytinyağında tipik olarak bulunan yağ asitleridir:

- **Palmitik Asit:** Doymuş yağ asididir, karbon zincirinde çift bağ bulunmaz. Zeytinyağındaki toplam yağ asidi içeriğinin yaklaşık %7.5-20'sini oluşturur.
- **Palmitoleik Asit:** Tekli doymamış yağ asididir, toplam yağ asidi içeriğinin yaklaşık %0.3-3.5'ini oluşturur.
- **Stearik Asit:** Doymuş yağ asididir, toplam yağ asidi içeriğinin yaklaşık %0.5-5'ini oluşturur.

- Oleik Asit: Zeytinyağındaki en bol bulunan yağ asididir, toplam yağ asidi içeriğinin yaklaşık %55-83'ünü oluşturur. Tekli doymamış yağ asididir.
- Linoleik Asit: Çoklu doymamış yağ asididir, karbon zincirinde iki veya daha fazla çift bağ bulunur. Toplam yağ asidi içeriğinin yaklaşık %3-21'ini oluşturur.
- Linolenik Asit: Başka bir çoklu doymamış yağ asididir, toplam yağ asidi içeriğinin yaklaşık %0.2-1.5'ini oluşturur.
- Arachidik Asit: Doymuş yağ asididir, toplam yağ asidi içeriğinin %1'inden azını oluşturur.
- Eikosenoik Asit: Tekli doymamış yağ asididir, toplam yağ asidi içeriğinin %1'inden azını oluşturur

Kod İncelenmesi:

Imports:



```
[14]: import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.mixture import GaussianMixture
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import cross_val_score, train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import svm
import sklearn.cluster
from sklearn.metrics import accuracy_score
```

Projede veri analizi, görselleştirme ve makine öğrenmesi modellerinin uygulanması için kullanılan Python kütüphaneleri ve fonksiyonlar aşağıda listelenmiştir:

- pandas: Veri işleme ve analiz.
- numpy: Matematiksel işlemler ve diziler.

- math: Matematiksel işlemler.
- matplotlib.pyplot: Grafikler ve görselleştirme.
- seaborn: İleri düzey görselleştirme.
- sklearn.cluster: Kümeleme algoritmaları.
- sklearn.mixture.GaussianMixture: Gaussian Mixture Model ile sınıflandırma.
- sklearn.preprocessing.LabelEncoder: Kategorik verilerin etiketlenmesi.
- sklearn.model_selection.cross_val_score, train_test_split: Model değerlendirme ve veri setinin bölünmesi.
- sklearn.tree.DecisionTreeClassifier: Karar ağacı sınıflandırıcısı.
- sklearn.svm: Destek vektör makineleri.
- sklearn.metrics.accuracy_score: Model doğruluğunu hesaplama.

Dataseti yüklenemesi:

[2]:

```
olive_data=pd.read_csv('C:\\Users\\Farhan Ahmad\\Desktop\\data mining\\olive.csv')
```

Verisetin Toplam Satır Sayısını Öğrenmek için:

•[26]:

```
print(len(olive_data))
```

618

Veri setinin Boyutu Öğrenmek İçin:

[4]:

```
olive_data.shape
```

[4]:

(572, 12)

Veri setinin içeriği görmek için:

[32]: `olive_data.head(12)`

	Region	palmitic	palmitoleic	stearic	oleic	linoleic	linolenic	arachidic	eicosenoic
0	organic extra virgin olive oil	2.0	1315.0	139.0	230.0	7299.0	832.0	42.0	60.0
1	organic extra virgin olive oil	2.0	1321.0	136.0	217.0	7174.0	950.0	43.0	63.0
2	organic extra virgin olive oil	2.0	1359.0	115.0	246.0	7234.0	874.0	45.0	63.0
3	organic extra virgin olive oil	2.0	1378.0	111.0	272.0	7127.0	940.0	46.0	64.0
4	organic extra virgin olive oil	2.0	1295.0	109.0	245.0	7253.0	903.0	43.0	62.0
5	organic extra virgin olive oil	2.0	1275.0	121.0	215.0	7285.0	892.0	40.0	68.0
6	organic extra virgin olive oil	2.0	1336.0	120.0	318.0	7083.0	915.0	50.0	70.0
7	organic extra virgin olive oil	2.0	1309.0	122.0	241.0	7257.0	870.0	46.0	72.0
8	organic extra virgin olive oil	2.0	1340.0	114.0	189.0	7337.0	820.0	48.0	72.0
9	organic extra virgin olive oil	2.0	1299.0	116.0	253.0	7309.0	823.0	40.0	69.0
10	organic extra virgin olive oil	2.0	1221.0	107.0	221.0	7441.0	798.0	54.0	70.0
11	organic extra virgin olive oil	2.0	1245.0	72.0	283.0	7395.0	829.0	44.0	67.0

Veri Setinin istatistiksel özetini öğrenmek için:

[31]: `olive_data.describe()`

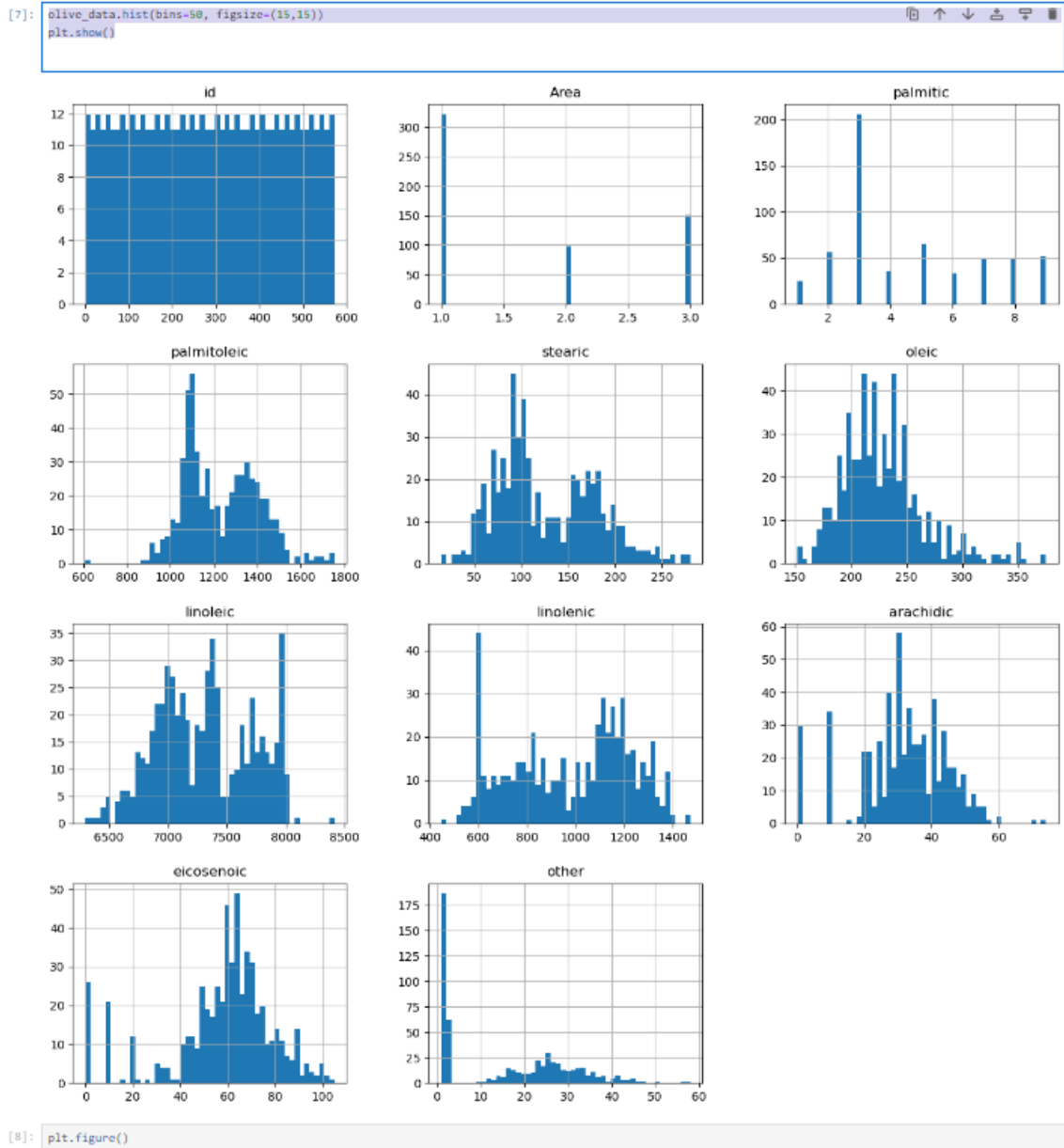
	palmitic	palmitoleic	stearic	oleic	linoleic	linolenic	arachidic	eicosenoic
count	618.000000	618.000000	618.000000	618.000000	618.000000	618.000000	618.000000	618.000000
mean	4.449504	1233.562036	124.921673	231.804911	7317.111866	966.461780	32.771522	58.736640
std	2.341121	164.916491	51.461328	38.163226	394.882957	240.719002	12.946103	21.458518
min	0.806950	610.000000	15.000000	152.000000	6300.000000	448.000000	0.000000	0.000000
25%	3.000000	1100.000000	88.000000	205.000000	7007.250000	760.361147	27.000000	51.144302
50%	3.000000	1213.500000	110.000000	226.500000	7320.000000	987.000000	34.000000	62.000000
75%	6.000000	1355.792620	166.000000	250.000000	7647.750000	1171.000000	42.000000	70.478675
max	9.000000	1753.000000	280.000000	375.000000	8410.000000	1470.000000	74.000000	105.000000

Veri setini histogramlarını çizmek için:

```
olive_data.hist(bins=50, figsize=(15,15))
```

```
plt.show()
```

hist fonksiyonuyla, veri kümenizdeki her bir sütun için bir histogram oluşturuyor. **bins** argümanı, histogramın çubuk sayısını belirtiyor **ve** **figsize** argümanı ise çizimin boyutunu beliriyor. **plt.show()** fonksiyonu ise çizimin ekranda gösterilmesini sağlıyor.

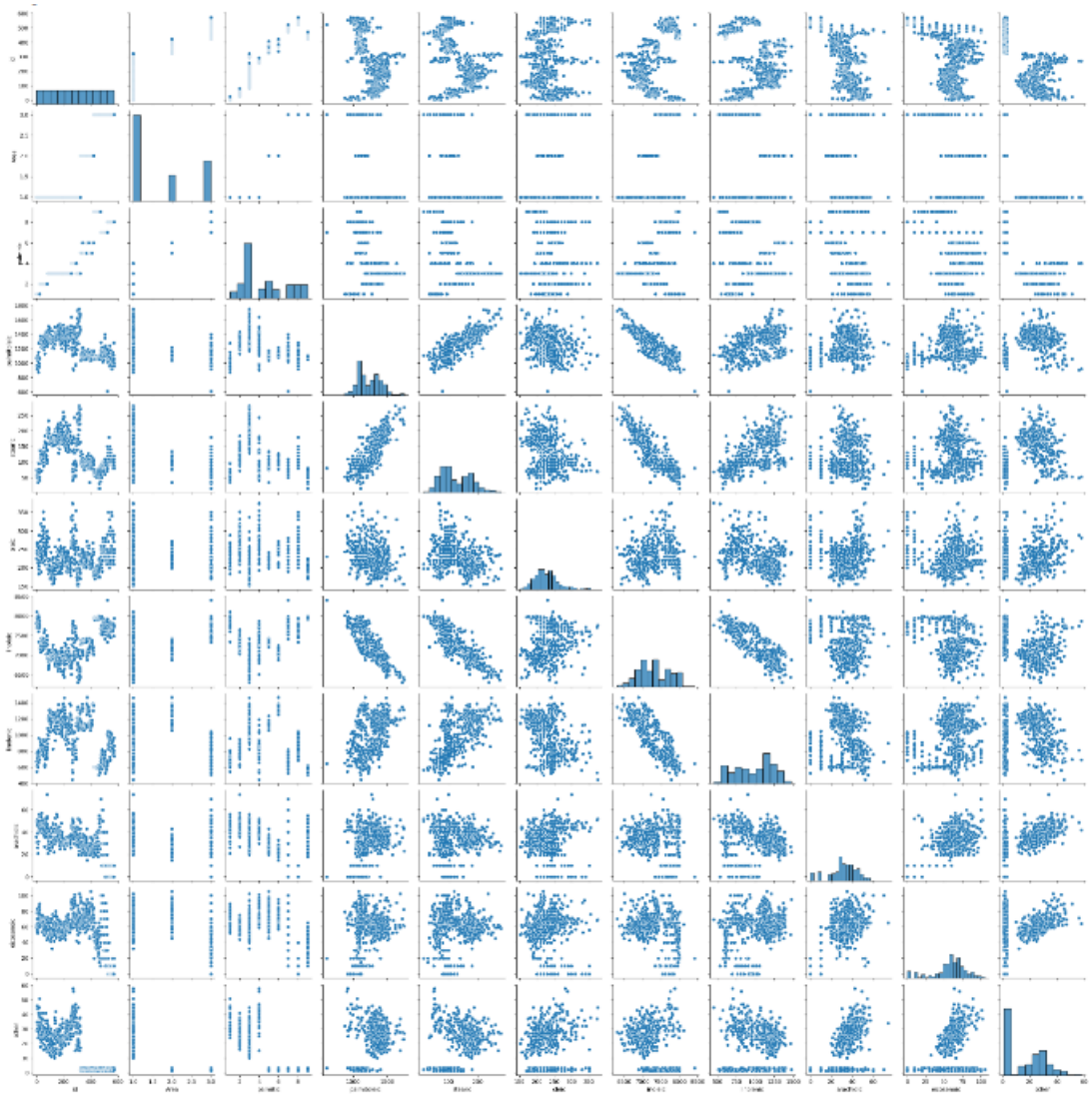


```
plt.figure()
```

```
sns.pairplot(olive_data, diag_kind='hist');
```

olive_data veri kümenizdeki sütunlar arasındaki ilişkiyi görselleştirmek için bir çift plot oluşturuyor. sns.pairplot fonksiyonu, veri kümenizdeki her bir sütun çifti için bir scatter plot çizer ve aynı zamanda her sütunun dağılımını gösteren bir histogram çizer.

diag_kind='hist' argümanı, diyagonaldeki grafiklerin histogram olmasını sağlar.



"Region" sütununda belirli bölgelerin değiştirilmiş değerlerini göstermek:

olive_data veri kümesindeki "Region" sütunundaki bazı değerleri değiştirir ve ardından bu sütundaki benzersiz değerleri yazdırır. Bu değişiklikler, belirli bölgelerin ya da yörelerin extra virgin olive oil veya organic extra virgin olive oil olarak sınıflandırılmasını sağlar. Son olarak, güncellenmiş veri kümesinde kaç örnek olduğunu ve "Region" sütununda hangi benzersiz değerlerin bulunduğunu göstermek için len(olive_data) ve olive_data['Region'].unique() komutları kullanılır.

```
[9]: olive_data.loc[olive_data['Region'] == 'North-Apulia', 'Region'] = 'extra virgin olive oil'
olive_data.loc[olive_data['Region'] == 'South-Apulia', 'Region'] = 'extra virgin olive oil'
olive_data.loc[olive_data['Region'] == 'Calabria', 'Region'] = 'organic extra virgin olive oil'
olive_data.loc[olive_data['Region'] == 'Sicily', 'Region'] = 'organic extra virgin olive oil'
olive_data.loc[olive_data['Region'] == 'Inland-Sardinia', 'Region'] = 'extra virgin olive oil'
olive_data.loc[olive_data['Region'] == 'Coast-Sardinia', 'Region'] = 'extra virgin olive oil'
olive_data.loc[olive_data['Region'] == 'Umbria', 'Region'] = 'extra virgin olive oil'
olive_data.loc[olive_data['Region'] == 'East-Liguria', 'Region'] = 'extra virgin olive oil'
olive_data.loc[olive_data['Region'] == 'West-Liguria', 'Region'] = 'extra virgin olive oil'
print(len(olive_data))
print(olive_data['Region'].unique())
```

572
['extra virgin olive oil' 'organic extra virgin olive oil']

Güncellenmiş veri kümesinin uzunluğu da yazdırılmak için:

olive_data veri kümesinden "id", "other" ve "Area" sütunlarını çıkarır ve ardından ilk iki satırını ve güncellenmiş veri kümesinin uzunluğunu yazdırır. drop fonksiyonu, belirtilen sütunu veri kümesinden çıkarır.

```
[10]: olive_data=olive_data.drop('id', axis=1)
olive_data=olive_data.drop('other', axis=1)
olive_data=olive_data.drop('Area', axis=1)
olive_data.head(2)
len(olive_data)
```

[10]: 572

Veri kümesindeki organik ekstra sızma zeytinyağı örneklerini artırmak için:

Gaussian karışım modelini (GaussianMixture) veriye uydurur. Veri olarak olive_data_organic veri kümesini kullanır. Model, belirtilen sayıda bileşene ayırır (n_components=1 ile yalnızca 1 bileşen seçilmiştir). Daha sonra, model kullanılarak veri seti üzerinde belirtilen sayıda örnek (n_samples) üretilir ve bu örnekler oluşturulan veri kümesine (generated_data) eklenir.

Son olarak, oluşturulan veri kümesi ile orijinal veri kümesi (olive_data_organic) birleştirilir (pd.concat). Bu işlem, orijinal veri kümesine eklenen yeni veri örneklerini içeren bir veri kümesi oluşturur. Oluşturulan veri örneklerinin "Region" sütununun değeri 'organic extra virgin olive oil' olarak ayarlanır.

```
[16]: # Fit the Gaussian mixture model to the data :
from sklearn.mixture import GaussianMixture
column_names = olive_data_organic.columns.tolist()

gmm = GaussianMixture(n_components=1) # we chose to have 1 container
gmm.fit(olive_data_organic)

n_samples = len(olive_data_organic) / 2
generated_data = gmm.sample(n_samples)[0]
generated_data = pd.DataFrame(generated_data, columns=column_names)

# concatenate the generated data with your original dataset
olive_data_organic = pd.concat([olive_data_organic, generated_data])

olive_data_organic['Region'] = 'organic extra virgin olive oil'
```

Oluşturduğunuz sentetik veri ile gerçek veriler bir araya getirmek için:

olive_data_organic ve olive_data_extra veri kümelerini birleştirir (pd.concat). Daha sonra, reset_index(drop=True) yöntemi kullanılarak indeks sıfırlanır ve birleştirilmiş veri kümesi (olive_data) yeniden düzenlenir.

```
[17]: # Reconcatenate the data :
olive_data = pd.concat([olive_data_organic, olive_data_extra])
olive_data = olive_data.reset_index(drop=True)
olive_data.head(2)
```

	Region	palmitic	palmitoleic	stearic	oleic	linoleic	linolenic	arachidic	eicosenoic
0	organic extra virgin olive oil	2.0	1315.0	139.0	230.0	7299.0	832.0	42.0	60.0
1	organic extra virgin olive oil	2.0	1321.0	136.0	217.0	7174.0	950.0	43.0	63.0

Özelliklerini ve hedef değişkene belirtmek:

veri kümenizi özellikler (X) ve hedef değişken (y) olarak ayırır. drop fonksiyonu ile 'Region' sütunu hedef değişken olarak seçilirken, geri kalan sütunlar özellikler olarak X değişkenine atanır.

Daha sonra, veri kümesi eğitim ve test setlerine ayrılır (train_test_split). Veri setinin yüzde 20'si test seti olarak ayrılırken, geri kalanı eğitim seti olarak kullanılır. random_state parametresi, veri setini rastgele bölerken kullanılacak olan rastgele durumun sabitlenmesini sağlar, bu da sonuçların tekrarlanabilir olmasını sağlar.

Son olarak, model_precision adında bir veri çerçevesi oluşturulur. Bu veri çerçevesi, modelin sınıflandırma doğruluğunu (precision) saklamak için kullanılacaktır. Her bir modelin doğruluğu bu veri çerçevesine eklenecektir.

```
[18]: # specify your features and target variable
X = olive_data.drop('Region', axis=1)
y = olive_data['Region']

# split the dataset into a training set and a testing set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model_precision = pd.DataFrame(columns=['Algorithm', 'precision'])
```

Organik ve Ekstra Sızma Zeytinyağı Veri Seti Ayırma İşlemleri:

Organic ve extra olarak etiketlenmiş verileri eğitim ve test veri setlerine ayırmak için kullanılıyor. İlk olarak, organic için eğitim ve test veri setleri oluşturuluyor. Eğitim veri setinin boyutu, olive_data_organic veri setinin üçte ikisi olarak belirleniyor. Ardından, test veri setinin boyutu hesaplanıyor ve eğitim veri setinden rastgele örnekler seçilerek oluşturuluyor. Aynı işlemler extra veri seti için de gerçekleştiriliyor. Son olarak, eğitim ve test veri setleri birleştirilerek model eğitimi ve değerlendirmesi için hazır hale getiriliyor.

```
#organic
nb_organic_train = math.floor((2 * len(olive_data_organic)) / 3)
nb_organic_test = 1 - nb_organic_train
olive_data_organic_train = olive_data_organic.sample(nb_organic_train)
random_indices_organic = olive_data_organic_train.index
olive_data_organic_test = olive_data_organic.drop(index=random_indices_organic)
olive_data_organic_train = olive_data_organic_train.reset_index(drop=True)
olive_data_organic_test = olive_data_organic_test.reset_index(drop=True)
olive_data_organic_train_features = olive_data_organic_train.copy()
olive_data_organic_train_features.drop('Region', axis=1)

olive_data_organic_test_features = olive_data_organic_test.copy()
olive_data_organic_test_features.drop('Region', axis=1)

olive_data_organic_train_Y = olive_data_organic_train_features['Region']
olive_data_organic_test_Y = olive_data_organic_test_features['Region']

#extra
nb_extra_train = math.floor((2 * len(olive_data_extra)) / 3)
nb_extra_test = 1 - nb_extra_train
olive_data_extra_train = olive_data_extra.sample(nb_extra_train)
random_indices_extra = olive_data_extra_train.index
olive_data_extra_test = olive_data_extra.drop(index=random_indices_extra)
olive_data_extra_train = olive_data_extra_train.reset_index(drop=True)
olive_data_extra_test = olive_data_extra_test.reset_index(drop=True)

olive_data_extra_train_features = olive_data_extra_train.copy()
olive_data_extra_train_features.drop('Region', axis=1)

olive_data_extra_test_features = olive_data_extra_test.copy()
olive_data_extra_test_features.drop('Region', axis=1)

olive_data_extra_train_Y = olive_data_extra_train_features['Region']
olive_data_extra_test_Y = olive_data_extra_test_features['Region']

olive_data_train_features = pd.concat([olive_data_extra_train_features, olive_data_organic_train_features])
olive_data_train_Y = pd.concat([olive_data_extra_train_Y, olive_data_organic_train_Y])

olive_data_test_features = pd.concat([olive_data_extra_test_features, olive_data_organic_test_features])
olive_data_test_Y = pd.concat([olive_data_extra_test_Y, olive_data_organic_test_Y])
```

Karar Ağacı Sınıflandırıcısı ile Zeytinyağı Sınıflandırma Doğruluğu Analizi:

Karar ağacı sınıflandırıcısı (DecisionTreeClassifier) kullanılarak bir model eğitiliyor ve test ediliyor. Modelin doğruluğu (accuracy) hesaplanıyor ve daha sonra bu doğruluk değeri, model_precision adlı bir DataFrame'e ekleniyor. Bu DataFrame'e her bir modelin sınıflandırma doğruluğu (precision) değerleri kaydedilerek, sonunda modellerin performansı karşılaştırılabilecek bir tablo oluşturulmuş olacak.

```
[23]: dt = DecisionTreeClassifier(random_state=42)
      dt.fit(X_train, y_train)
      y_pred = dt.predict(X_test)
      accuracy = accuracy_score(y_test, y_pred)
      print("Accuracy:", accuracy)

      # Assuming model_precision is already defined as a DataFrame
      model_precision = model_precision.append({'Algorithm': 'Decision Tree', 'precision': accuracy}, ignore_index=True)

      Accuracy: 0.9516129032258065
```

Anlattım video

<https://www.youtube.com/playlist?list=PLNe57ElYnJlCqKYBjkwbmGYl2Ber9o0t>

Kaynak:

<https://gcris.iyte.edu.tr/handle/11147/13276>

https://www.researchgate.net/publication/5269125_Classification_of_extra_virgin_olive_oils_according_to_the_protected_designation_of_origin_olive_variety_and_geographical_origin

<https://www.sciencedirect.com/science/article/abs/pii/S0308814623014115>

<https://pubmed.ncbi.nlm.nih.gov/30625602/>

<https://www.mdpi.com/1420-3049/28/3/1483>