-VERİ MADENCİLİĞİ RAPOR-

Hazırlayan: Ali Doran
Video Linki:https://youtu.be/OIWwccXVuWA
GitHub Linki:https://github.com/doranali/Veri_Madencili-i_Wine.git

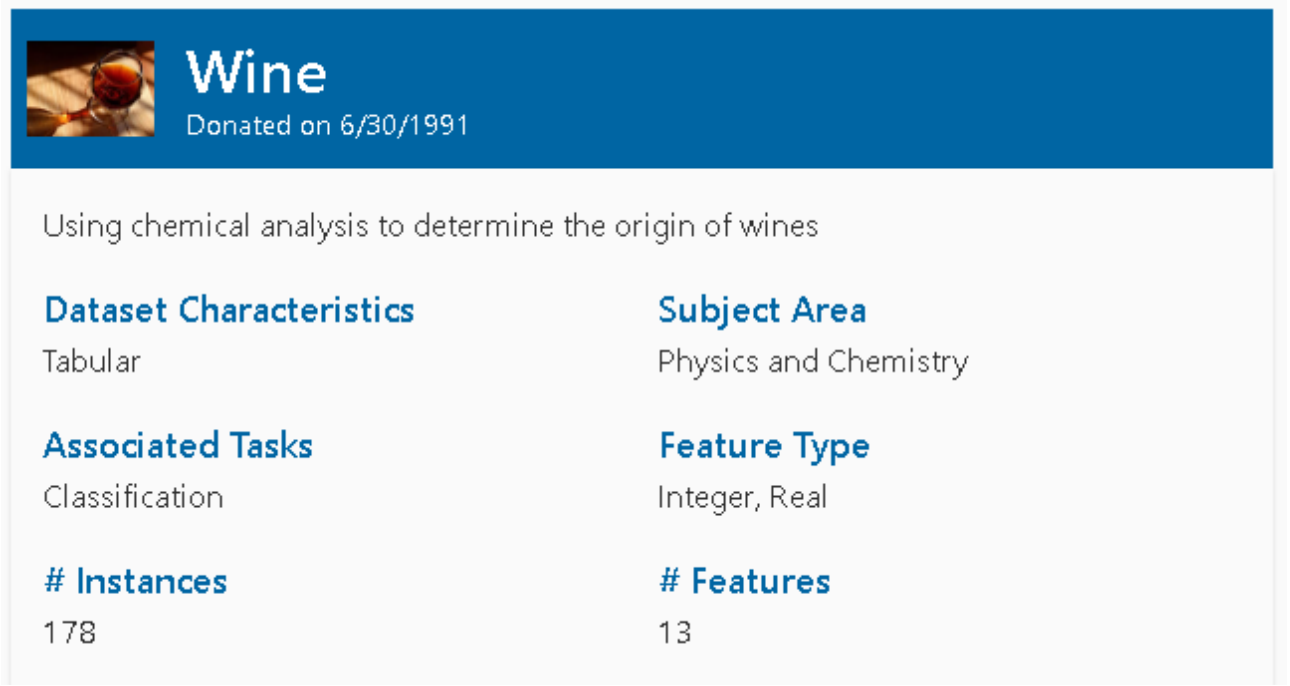
Karar Ağacı Yöntemi ile Wine Veri Seti Üzerinden Sınıflandırma ve Model Karşılaştırması

1. Projenin Amacı ve Kullanılan Veri Seti

Bu projede, karar ağacı algoritması kullanarak çok sınıflı bir veri seti üzerinde sınıflandırma işlemi gerçekleştirmeyi hedefledim. Asıl amacım, bir modelin ham veriyle nasıl performans gösterdiğini görmek ve ardından öznitelik seçimi ile budama işlemleri gibi iyileştirmeleri uygulayarak modelin başarımındaki değişimi değerlendirmekti. Bu süreci gerçekleştirirken sadece sonuçlara değil, sürecin tüm aşamalarına odaklanarak modelin daha anlaşılır ve yorumlanabilir hale gelmesini amaçladım.

Kullanılan Veri Seti

Çalışmamda UCI Machine Learning Repository'den alınan klasik “Wine” veri setini kullandım. Bu veri seti, farklı şarap türlerine ait 13 kimyasal özellikten oluşmaktadır. Hedef değişken (class), bu şarapların ait olduğu üç farklı sınıfı belirtmektedir. Toplamda 178 gözlemden oluşan bu veri seti, sınıflandırma problemleri için oldukça elverişli bir yapıya sahiptir.



Çalışmanın Ana Aşamaları

1. İlk olarak, ham veri üzerinde hiçbir ön işleme yapmadan karar ağacı modeli eğittim ve temel metriklerle performansını analiz ettim.
2. Daha sonra, aynı veri setine öznitelik seçimi ve budama işlemleri uygulayarak daha sade ve genellenebilir bir model oluşturdum.
3. Elde edilen bu iyileştirilmiş model ile ilk modeli karşılaştırarak, yapılan işlemlerin model başarısına olan etkisini değerlendirdim.
4. Son olarak, Wine veri seti ile daha önce yapılmış olan bir çalışmayı inceleyerek kendi modelimle nicel bir kıyaslama gerçekleştirdim.

Bu proje, ham veriden başlayarak iyileştirilmiş modele geçiş sürecini adım adım izlemeyi ve her aşamada karar ağaçlarıyla sınıflandırma sürecinin hem uygulamalı hem de analitik olarak değerlendirilmesini amaçlamaktadır.

2. Veri Setinin Eğitim ve Test Olarak Ayrılması

Modelleme sürecine başlamadan önce, veri setini eğitim ve test olmak üzere ikiye ayırdım. Bu ayrımın temel amacı, oluşturduğum karar ağacı modelinin yalnızca eğitildiği veriler üzerinde değil, daha önce hiç görmediği örnekler üzerinde de test edilmesini sağlayarak genelleme performansını objektif biçimde değerlendirebilmekti.

Veri bölme işlemini Python ortamında train_test_split fonksiyonu yardımıyla gerçekleştirdim. Eğitim verisi %70, test verisi ise %30 oranında belirlendi. Ayrıca, sınıflar arasında dengesizlik oluşmaması için stratify parametresini kullanarak bölünmenin Class değişkenine göre dengeli olmasını sağladım.

Aşağıda kullandığım kod parçası yer almaktadır:

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Veri setini oku
df = pd.read_csv("wine.csv")

# Eğitim ve test verilerini %70 - %30 oranında ayır (Class
sütununa göre)
train_df, test_df = train_test_split(df, test_size=0.3,
random_state=42, stratify=df['Class'])

# Ayırdığın dosyaları kaydet
train_df.to_csv("wine_train.csv", index=False)
test_df.to_csv("wine_test.csv", index=False)

print("Veri başarıyla ayrıldı ve kaydedildi!")
```

Bu işlem sonucunda, bir yandan modelin öğrenmesi için gerekli olan eğitim verisi elde edilirken, diğer yandan modelin başarımını tarafsız şekilde değerlendirebileceğim bir test veri kümesi oluşturulmuş oldu. Böylece, sonraki adımlarda yapacağım modelleme ve karşılaştırmalarda elde ettiğim sonuçların daha güvenilir olmasını sağlamayı hedefledim.

3. Kullanılan Teknolojiler ve Uygulama Ortamı

Bu projede karar ağacı tabanlı sınıflandırma modelinin geliştirilmesi, eğitimi ve değerlendirilmesi süreçlerinde Python programlama dilinden faydalandım. Kodlama ve analiz işlemlerini Jupyter Notebook ortamında gerçekleştirdim. Bu sayede, tüm işlem adımlarını hem interaktif olarak hem de adım adım takip edilebilir ve yeniden üretilebilir biçimde yürütme imkânı elde ettim.

Veri işleme, modelleme ve görselleştirme işlemleri için aşağıdaki Python kütüphanelerinden yararlandım:

- **pandas**: Veri dosyalarını okuma, yazma ve tablo bazlı veri manipülasyon işlemleri için kullanılmıştır.
- **scikit-learn (sklearn)**: Karar ağacı modelinin oluşturulması, öznitelik seçimi, model eğitimi ve değerlendirme metriklerinin hesaplanmasında temel rol oynamıştır.
- **matplotlib & seaborn**: Verilerin görsel olarak analiz edilmesinde ve karşılaştırmalı grafiklerin hazırlanmasında kullanılmıştır.
- **numpy**: Sayısal hesaplamalarda destekleyici temel yapı taşlarını sağlamıştır.
- **Orange**: Görsel tabanlı veri madenciliği aracı olarak, modelin anlaşılabilirlik düzeyini artırmak ve sonuçları sezgisel biçimde sunmak amacıyla kullanılmıştır.

Model eğitimi ve test aşamaları için gerekli olan veri dosyalarını projeye aşağıdaki kod bloğu ile dahil ettim:

```
import pandas as pd

# Eğitim ve test veri setlerini oku
train_df = pd.read_csv("wine_train.csv")
test_df = pd.read_csv("wine_test.csv")
```

Ardından, veri çerçevelerinden giriş (özellik) ve hedef (etiket) sütunlarını ayırarak eğitim ve test setlerini şu şekilde oluşturdum:

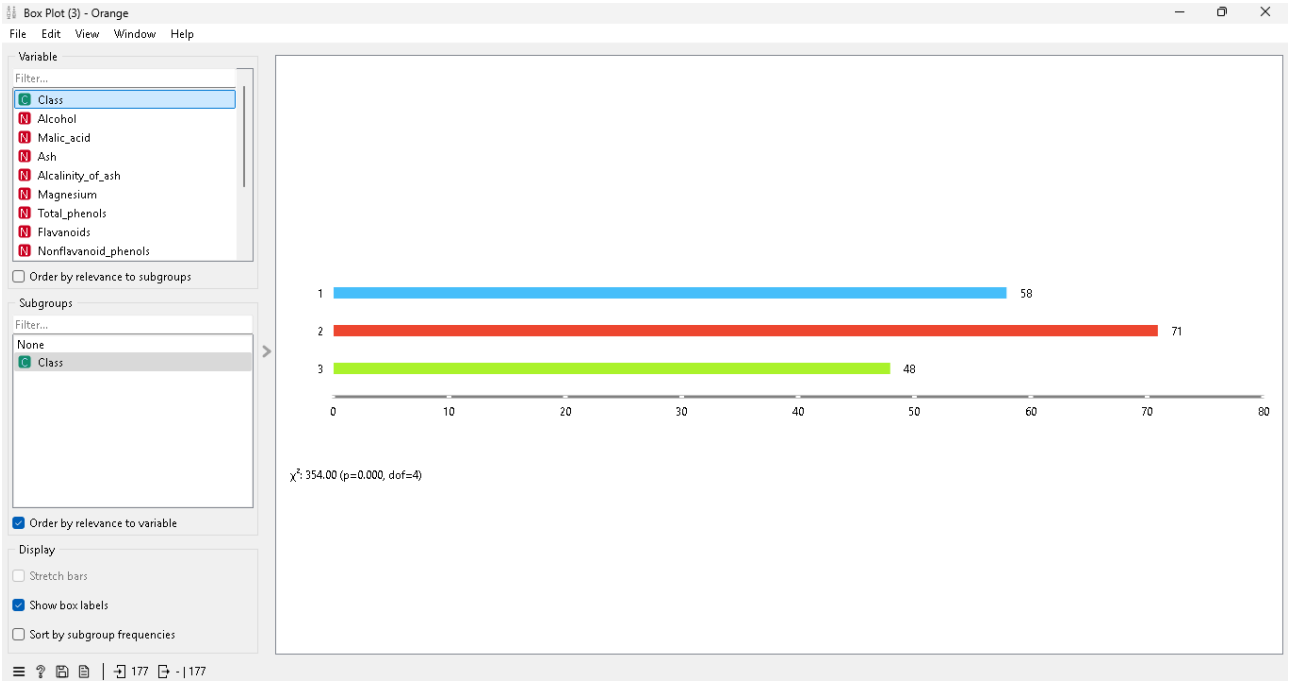
```
X_train = train_df.drop(columns=["Class"])
y_train = train_df["Class"]

X_test = test_df.drop(columns=["Class"])
y_test = test_df["Class"]
```

Bu ortam sayesinde model eğitimi ve değerlendirme süreçleri adım adım takip edilebilir ve yeniden üretilebilir biçimde yürütülmüştür.

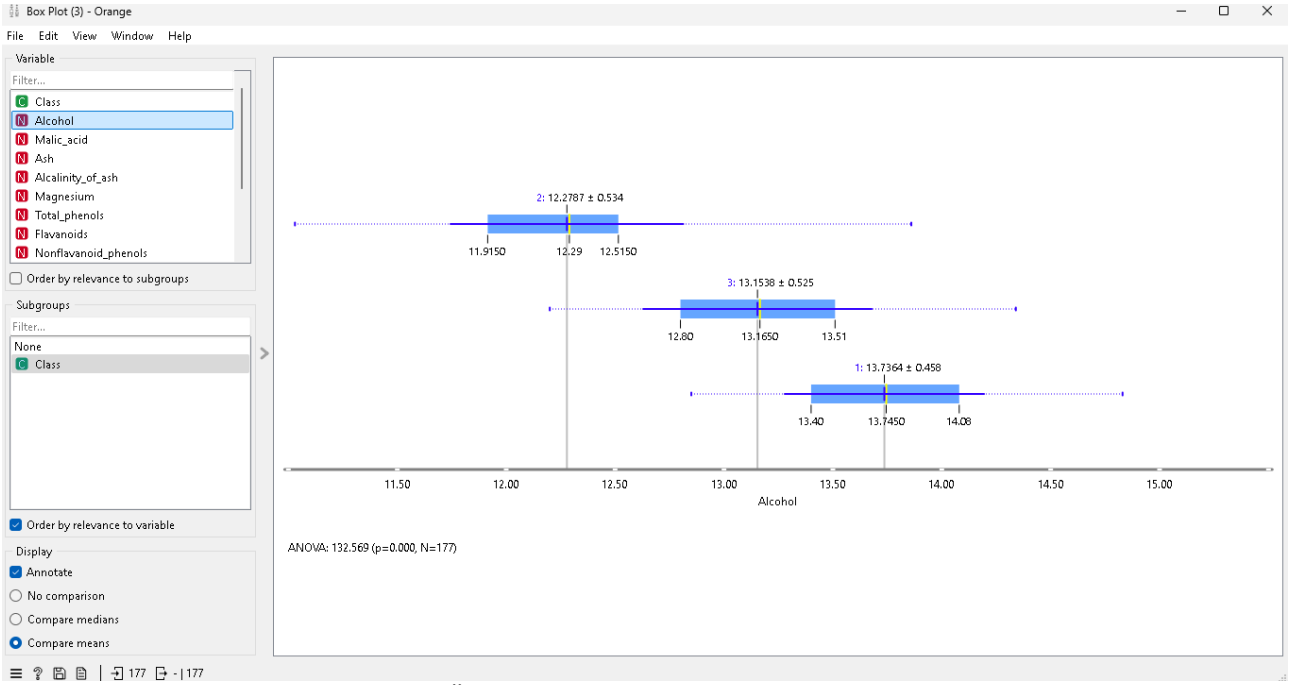
3. Ham Veri Üzerinde Görsel ve İstatistiksel Özellik Analizi

Modelleme adımlarına geçmeden önce, veri setinin ham hâli üzerinden bazı temel özniteliklerin sınıflara göre dağılımı görselleştirilmiş ve istatistiksel olarak incelenmiştir. Bu analiz, karar ağacı modeline girdi sağlayan özniteliklerin sınıflar üzerindeki etkisini daha iyi anlamak ve modelin hangi değişkenlere dayandığını daha sağlam temellere oturtmak amacıyla gerçekleştirilmiştir.
Bu bölümde yer alan grafikler, sınıfların (1, 2, 3) belirli nitelikler açısından ne derece ayrıştığını ortaya koymakta ve modelin karar mekanizmasında etkili olabilecek değişkenler hakkında ön bilgi sunmaktadır. Görselleştirmeler hem kutu grafikleri (box plot), hem saçılım grafikleri (scatter plot), hem de yoğunluk grafikleri (violin plot) biçiminde sunulmuştur.



Şekil 1: Sınıf Dağılımının Gözlemlendiği Çubuk Grafik

Bu grafikte, ham veri setinde yer alan örneklerin üç farklı şarap sınıfına göre dağılımı çubuk grafik biçiminde gösterilmiştir.
Grafikte gördüğüm üzere, en fazla sayıda örnek sınıf 2'ye (71 adet) aitken, sınıf 1'e 58 ve sınıf 3'e 48 örnek düşmektedir. Bu dağılım, sınıflar arası belirgin bir dengesizlik olmadığını göstermektedir. Ayrıca, grafiğin alt kısmında yer alan ki-kare testi sonucu ($\chi^2 = 354.00$, $p = 0.000$) da bu sınıf dağılımının istatistiksel olarak anlamlı şekilde farklılaştığını göstermektedir. Bu, veri setindeki sınıf bilgisiyle ilişkili özniteliklerin karar ağacı modeli tarafından daha etkili öğrenilmesine olanak sağlayabilir.

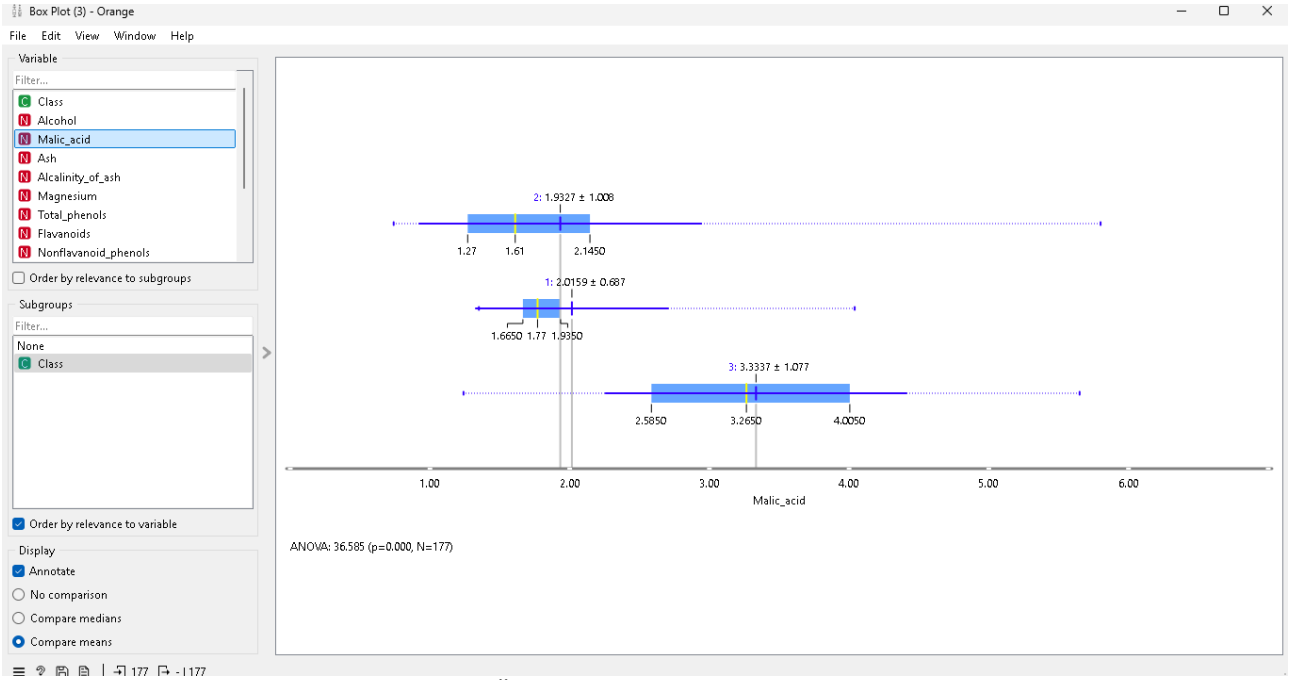


Şekil 2: Sınıflara Göre "Alcohol" Özelliğinin Dağılımı (Box Plot)

Bu kutu grafiği, "Alcohol" özniteliğinin üç sınıf özelinde dağılımını göstermektedir. Gözlemlerime göre:

- Sınıf 1 (mavi): Ortalama alkol değeri 13.7364 ± 0.458
- Sınıf 2 (kırmızı): Ortalama 12.2787 ± 0.534
- Sınıf 3 (yeşil): Ortalama 13.1538 ± 0.525

Aralarında anlamlı bir fark olduğunu ANOVA sonucu da desteklemektedir ($F = 132.569$, $p = 0.000$). Bu bulgular, "Alcohol" değişkeninin sınıflar arasında ayırt edici bir rol oynadığını ve özellikle karar ağacında üst düğümlerde kullanıldığında anlamlı sonuçlar vereceğini göstermektedir.

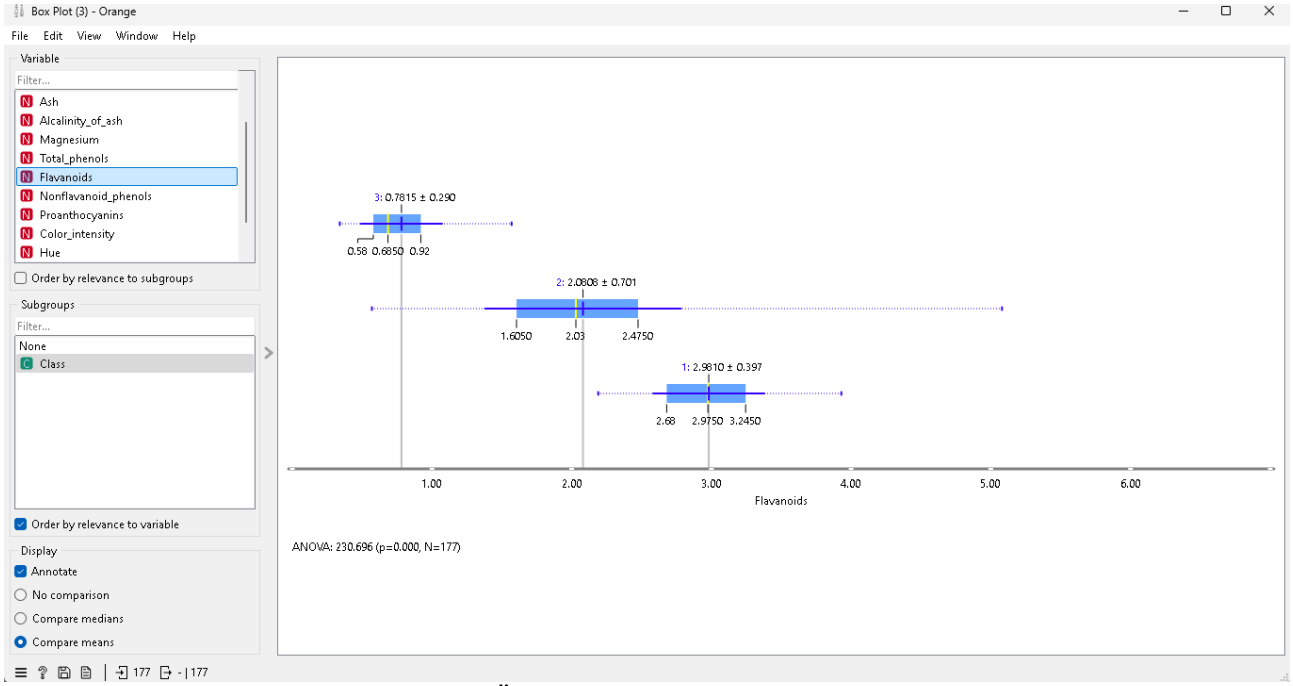


Şekil 3: Sınıflara Göre "Malic_acid" Özelliğinin Dağılımı (Box Plot)

"Malic_acid" özniteliğine ait bu kutu grafiğinde üç sınıf arasında dikkat çekici farklar gözlemledim:

- Sınıf 1: Ortalama 2.0159 ± 0.687
- Sınıf 2: Ortalama 1.9327 ± 1.008
- Sınıf 3: Ortalama 3.3337 ± 1.077

Özellikle sınıf 3'teki değerlerin belirgin biçimde daha yüksek olduğu açıkça görülmektedir. ANOVA testi (F = 36.585, p = 0.000) sonucunda elde edilen anlamlı fark, bu değişkenin sınıflar arası ayrımı destekleyen güçlü bir öznitelik olduğunu doğrulamaktadır.

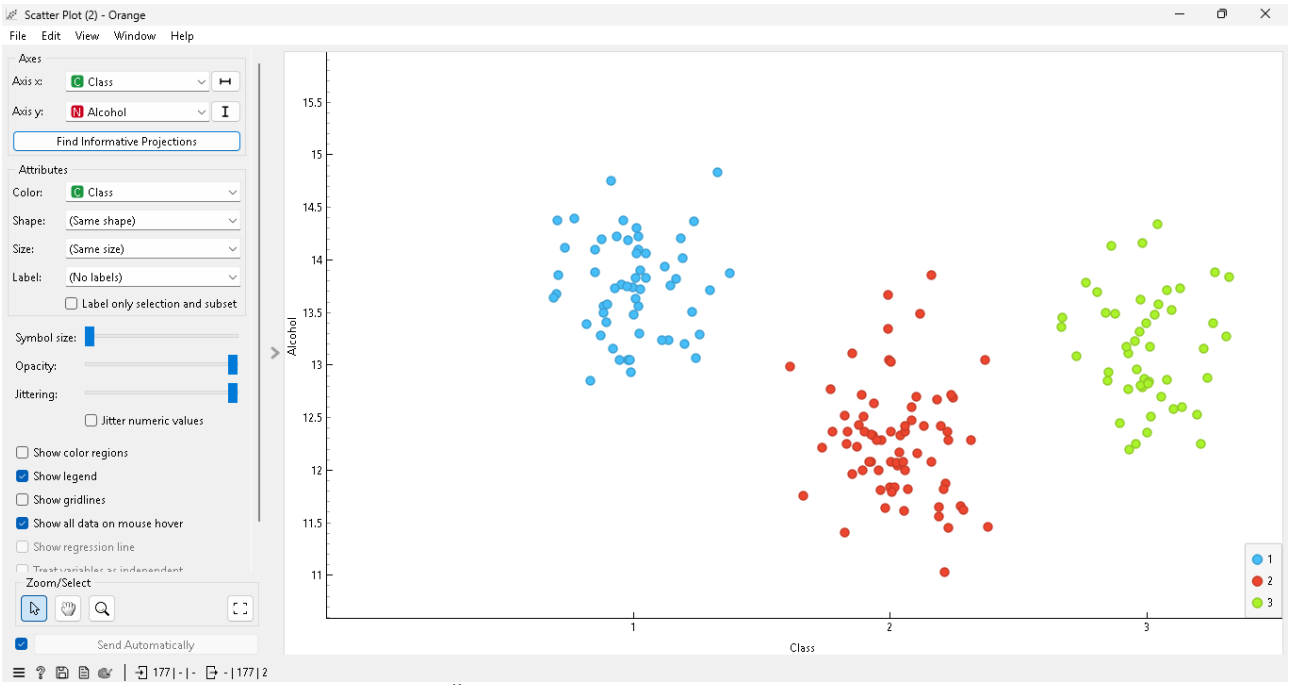


Şekil 4: Sınıflara Göre "Flavonoids" Özelliğinin Dağılımı (Box Plot)

Bu görselde "Flavonoids" özniteliğinin üç sınıf üzerindeki dağılımı istatistiksel olarak analiz edilmiştir.
Gördüğüm sonuçlar şu şekildedir:

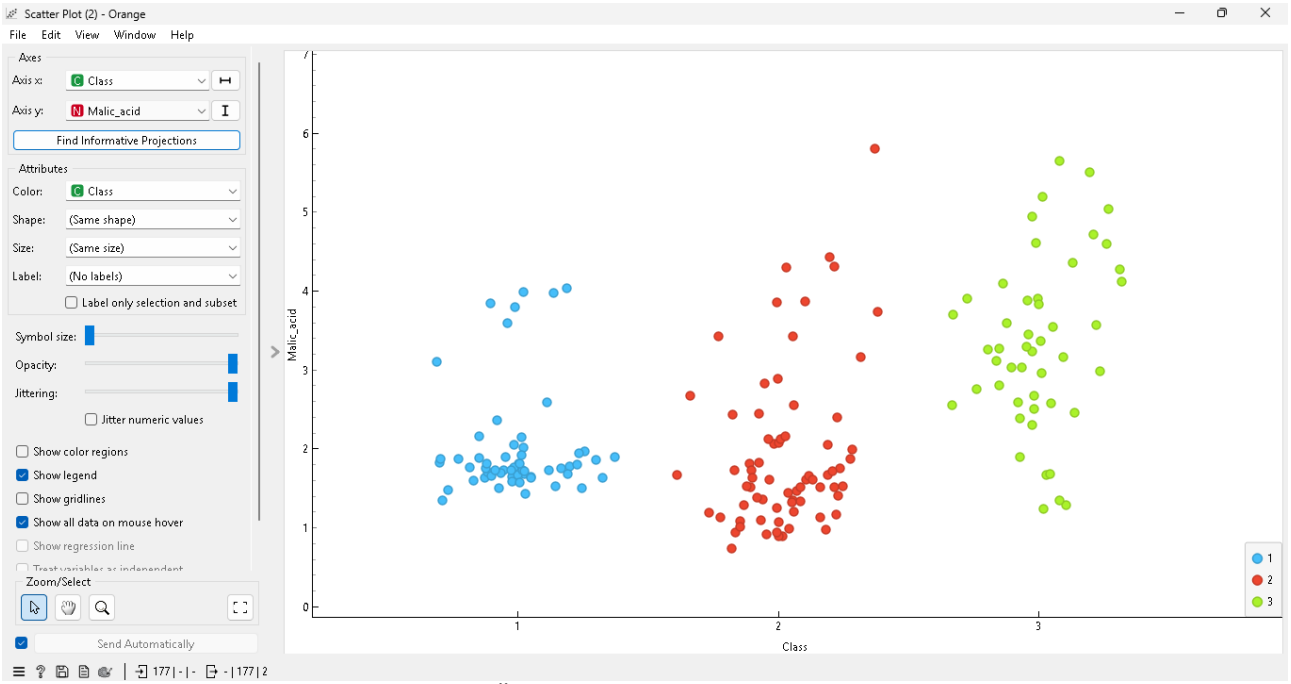
- Sınıf 1: 2.9810 ± 0.397
- Sınıf 2: 2.0608 ± 0.701
- Sınıf 3: 0.7815 ± 0.290

Sınıf 3'ün bu değişkene ait değerleri oldukça düşüktür ve sınıf 1 ile aralarında net bir fark bulunmaktadır.
ANOVA testi çıktısı (F = 230.696, p = 0.000) da bu değişkenin sınıflar arasında güçlü bir ayırt edici rol üstlendiğini istatistiksel olarak kanıtlamaktadır.
Modelin bu değişkene üst düğümlerde öncelik vermesi, gözlemsel olarak da oldukça tutarlı görünmektedir.



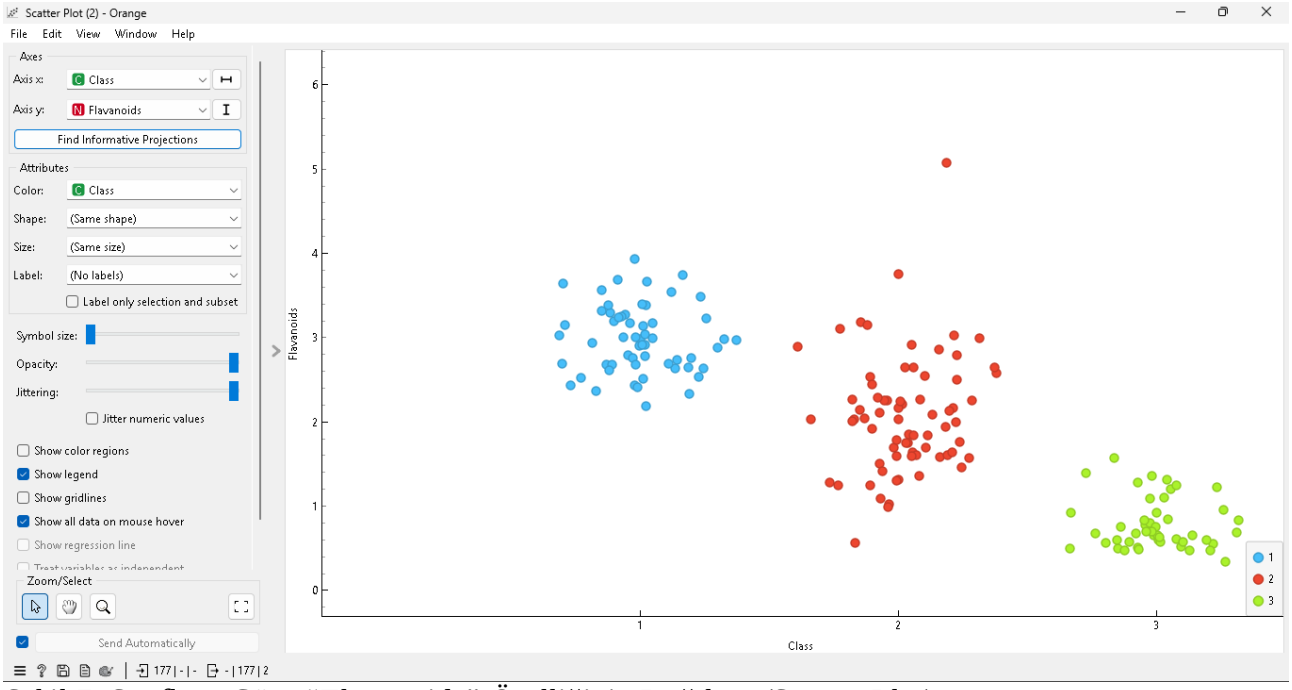
Şekil 5: Sınıflara Göre "Alcohol" Özelliğinin Dağılımı (Scatter Plot)

Bu saçılım grafiği, alkol değişkeninin sınıflar arasında nasıl bir dağılım gösterdiğini daha detaylı bir şekilde gözlemleyebilmemi sağladı. Sınıf 1 (mavi) bireylerinin yüksek alkol düzeylerinde yoğunlaştığını, sınıf 2 (kırmızı) bireylerinin daha düşük seviyelerde toplandığını ve sınıf 3 (yeşil) bireylerinin ise görece daha ortalamaya yakın bir yayılım sergilediğini fark ettim. Bu açık ayrışma, alkol değişkeninin sınıf tahmininde etkili bir belirleyici olduğunu ve karar ağacının ayrım yaparken bu özelliği kullanmasının anlamlı olduğunu göstermektedir.



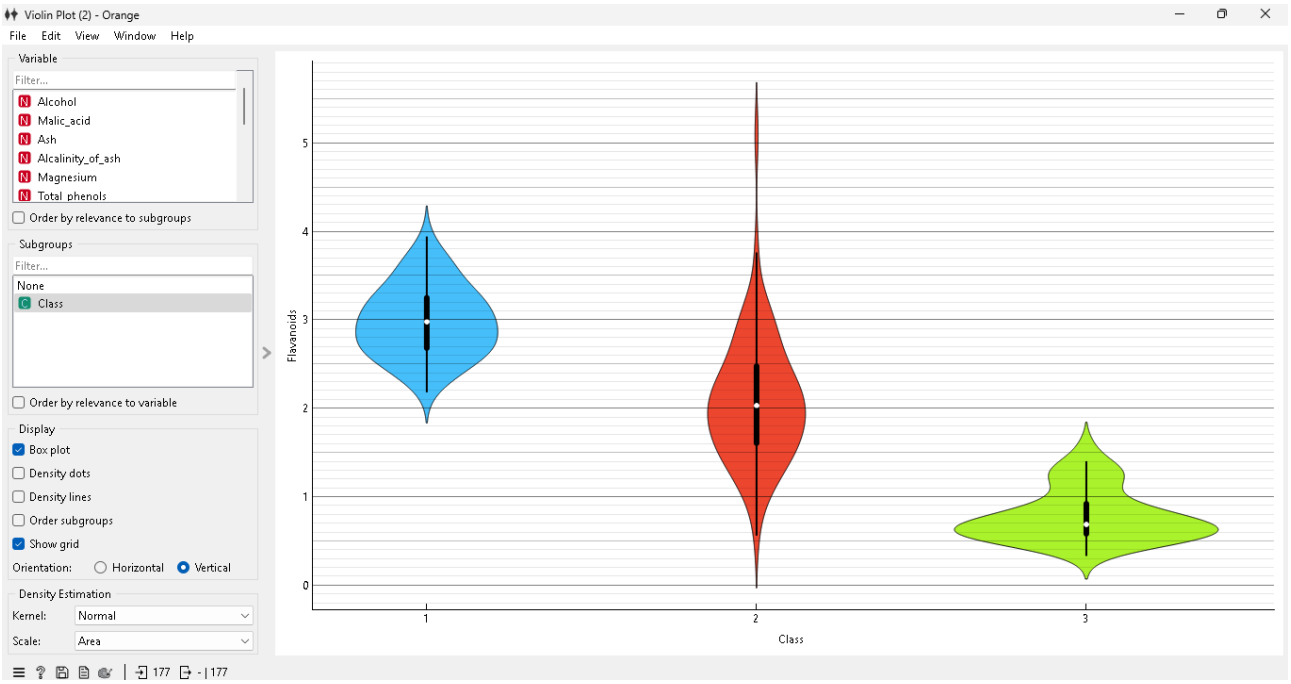
Şekil 6: Sınıflara Göre "Malic_acid" Özelliğinin Dağılımı (Scatter Plot)

Bu grafikte, Malic_acid değişkeninin üç sınıf üzerindeki yayılımını inceledim. Sınıf 1 (mavi) bireyleri düşük Malic_acid seviyelerinde toplanmışken, sınıf 2 (kırmızı) bireyleri orta seviyelerde, sınıf 3 (yeşil) bireyleri ise yüksek Malic_acid değerlerine sahip. Bu dağılım, bu özniteliğin sınıflar arası ayrımda güçlü bir gösterge olduğunu ortaya koymaktadır. Görseldeki kümelenme yapısı, bu değişkenin karar ağacı gibi sınıflandırıcı modellerde etkin bir biçimde kullanılabileceğini göstermektedir.



Şekil 7: Sınıflara Göre "Flavanoids" Özelliğinin Dağılımı (Scatter Plot)

Bu scatter plot, Flavanoids değişkeninin sınıflar arası dağılımını net bir şekilde ortaya koymaktadır. Sınıf 1 (mavi) bireyleri yüksek Flavanoids değerlerinde kümelenmişken, sınıf 2 (kırmızı) bireylerinde bu değişken orta seviyelerde dalgalanmakta, sınıf 3 (yeşil) bireylerinde ise oldukça düşük değerlere yoğunlaşmaktadır. Bu net ayrım, Flavanoids değişkeninin modelleme sürecindeki önemini doğrular niteliktedir ve karar ağacındaki üst düğümlerde yer almasını istatistiksel olarak desteklemektedir.



Şekil 8: "Flavanoids" Değişkeninin Sınıflara Göre Yoğunluk Dağılımı (Violin Plot)

Bu keman grafiği, Flavanoids değişkeninin sınıflar içindeki dağılım yoğunluğunu anlamam açısından oldukça faydalı oldu. Sınıf 1 (mavi) bireyleri nispeten simetrik ve yüksek yoğunluklu değerlere sahiptir. Sınıf 2 (kırmızı) bireylerinde daha geniş ve dengesiz bir dağılım gözlemlenirken, sınıf 3 (yeşil) bireylerinin çok daha düşük ve dar bir dağılım aralığında yer aldığı görülmektedir. Bu görsel, sınıflar arası varyasyonun yalnızca ortalamalarla değil, aynı zamanda dağılım yapısıyla da ayırt edici olduğunu açıkça göstermektedir.

Bu bölümde gerçekleştirdiğim görselleştirme ve istatistiksel analizler sayesinde, veri setinde yer alan bazı temel özniteliklerin sınıflar üzerinde anlamlı bir ayrım gücüne sahip olduğunu açıkça gözlemledim. Özellikle *Alcohol*, *Malic_acid* ve *Flavanoids* değişkenleri hem dağılımlarıyla hem de sınıflar arası istatistiksel farklılıklarıyla dikkat çekmiştir. Bu durum, bu özniteliklerin karar ağacı gibi ayrım temelli algoritmalarda etkili şekilde kullanılabileceğini göstermektedir.

Elde ettiğim bu bulgular, ilerleyen bölümlerde oluşturacağım karar ağacı modelinin temel öznitelik tercihlerine ilişkin ön bilgi sağlamıştır. Ayrıca, modelleme aşamasında gözlemlenen başarımın hangi değişkenler tarafından daha çok desteklenmiş olabileceğini değerlendirmeme de katkı sunacaktır. Bu analiz, ham veri ile oluşturulacak ilk modelin değerlendirilmesinde yorum gücümü artırmak amacıyla gerçekleştirilmiştir.

4. Model 1 – Ham Veri Üzerinde Budamasız Karar Ağacı Uygulaması

Modelleme sürecinin ilk adımında, veri seti üzerinde herhangi bir ön işleme, öznitelik seçimi veya budama işlemi uygulanmadan bir karar ağacı modeli oluşturdum. Bu modelin amacı, veriye hiçbir müdahale yapılmadan elde edilen temel sınıflandırma başarımını değerlendirmek ve bu sonuçları ileride uygulanacak tekniklerle karşılaştırmak için bir referans noktası oluşturmaktır.

Bu aşamada, tüm öznitelikler doğrudan modele dâhil edilmiştir. Model, DecisionTreeClassifier sınıfı kullanılarak varsayılan parametrelerle eğitilmiştir. Aşağıda ilgili Python kodu yer almaktadır:

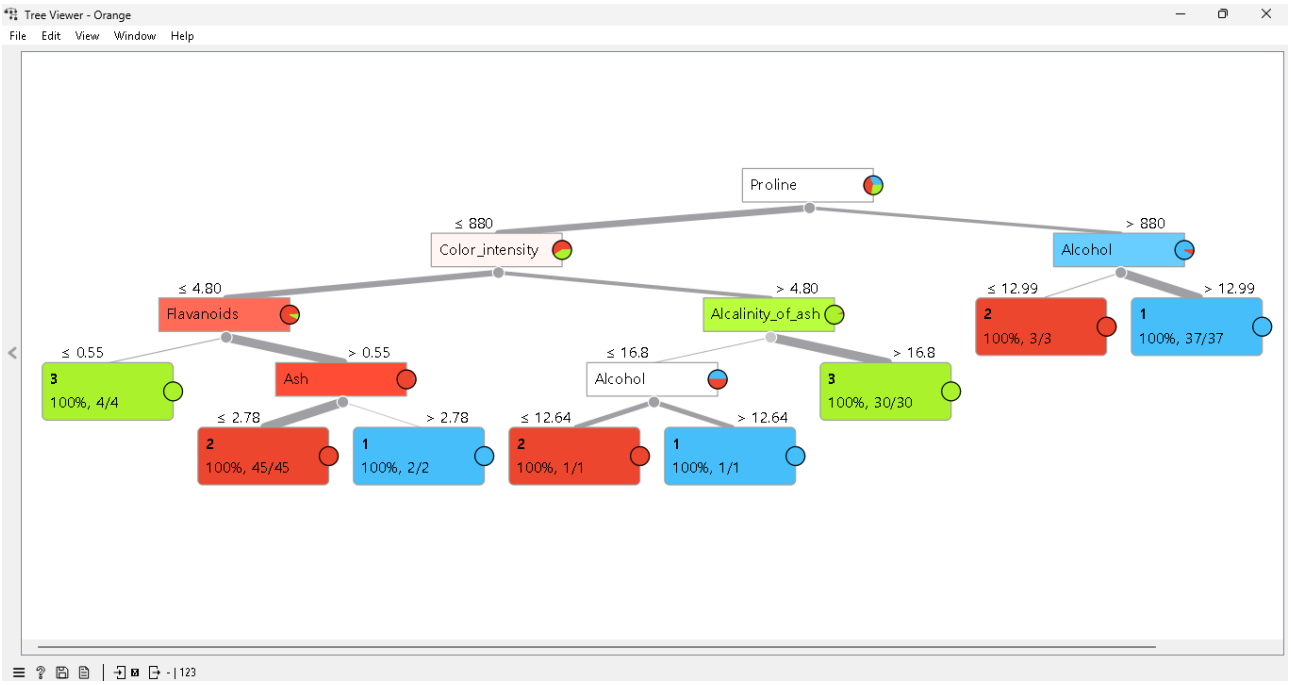
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, f1_score,
precision_score, recall_score, matthews_corrcoef, roc_auc_score
from sklearn.preprocessing import label_binarize

# Model tanımlama ve eğitme
model_base = DecisionTreeClassifier(random_state=42)
model_base.fit(X_train, y_train)

# Tahmin ve değerlendirme
y_pred_base = model_base.predict(X_test)
y_proba_base = model_base.predict_proba(X_test)

# AUC için çok sınıflı binarizasyon
y_test_bin = label_binarize(y_test, classes=[1, 2, 3])
auc_base = roc_auc_score(y_test_bin, y_proba_base,
multi_class="ovo")

# Performans metrikleri
print("Accuracy:", round(accuracy_score(y_test, y_pred_base), 3))
print("F1 Score:", round(f1_score(y_test, y_pred_base,
average="macro"), 3))
print("Precision:", round(precision_score(y_test, y_pred_base,
average="macro"), 3))
print("Recall:", round(recall_score(y_test, y_pred_base,
average="macro"), 3))
print("MCC:", round(matthews_corrcoef(y_test, y_pred_base), 3))
print("AUC:", round(auc_base, 3))
```



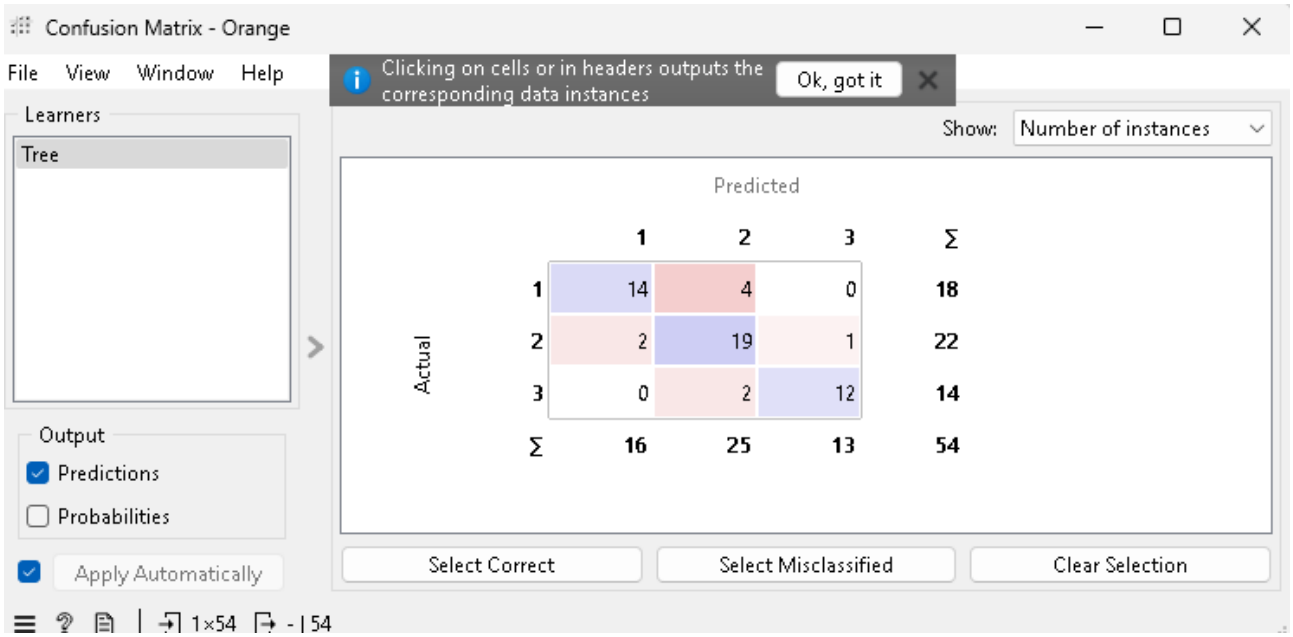
Şekil 9: Ham veri üzerinde eğitilen budamasız karar ağacı modelinin ağaç yapısı. Görselde sınıflar renkli olarak ayrılmış, her düğümde örnek sayısı ve sınıf dağılımı gösterilmiştir.

Model, tüm öznitelikleri kullanarak veriyi sınıflandırmaktadır. Bu durum modelin eğitimi açısından avantaj sağlasa da, karmaşık yapı nedeniyle aşırı öğrenmeye (overfitting) açık hâle gelmektedir. Özellikle eğitim verisinde yüksek başarı göstermesi beklenirken, test verisine genellenebilirliği sınırlı olabilir. Bu nedenle bu model, ileride yapılacak öznitelik seçimi ve budama işlemleriyle karşılaştırmak amacıyla temel bir karşılaştırma noktası olarak kullanılmıştır.

```
 [MODEL 1 - Ham Veri, Budamasız]
Accuracy: 0.815
F1 Score: 0.82
Precision: 0.841
Recall: 0.809
MCC: 0.719
AUC: 0.855
```

Şekil 10: Modelin test verisiyle başarımı

Bu metrikler genel anlamda tatmin edici düzeydedir ve modelin sınıflar arasında anlamlı ayrımlar yapabildiğini göstermektedir. Ancak daha sonra geliştirilecek modellerde bu performansın artırılması hedeflenmiştir.

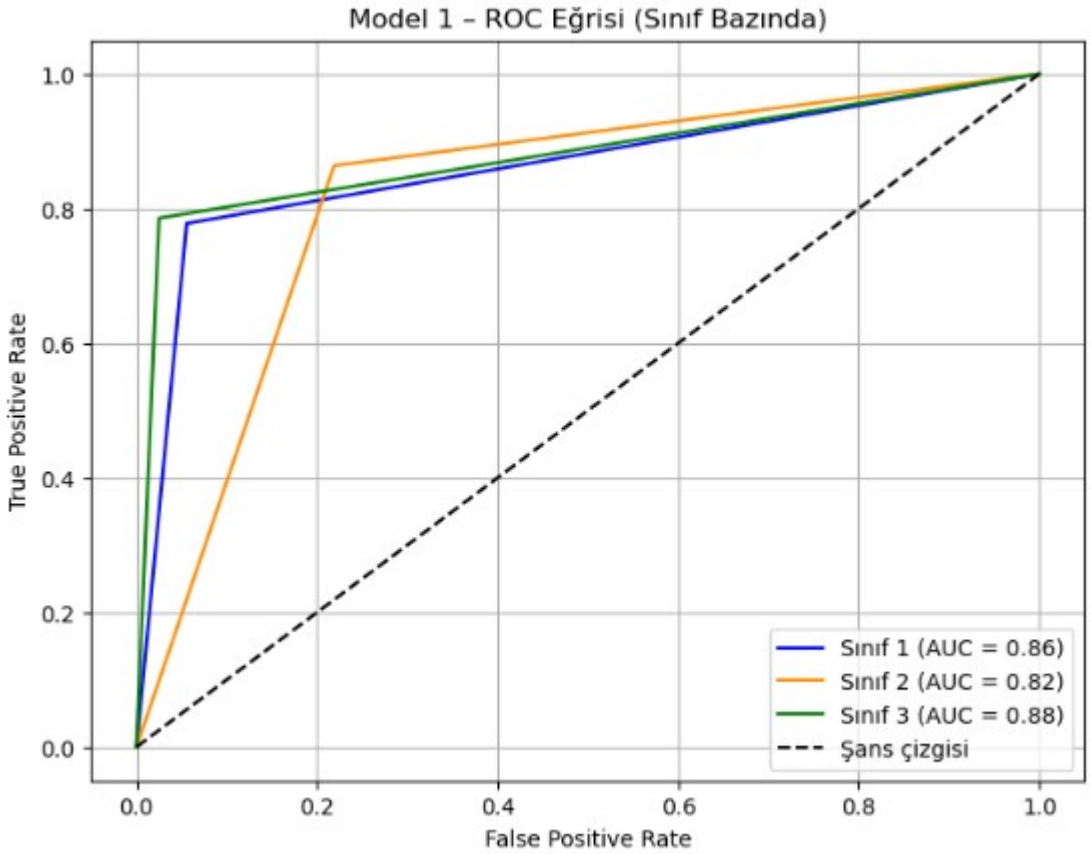


Şekil 11: Model 1'in test veri kümesi üzerindeki karışıklık matrisi. Her hücre, modelin tahmin ettiği ve gerçek sınıf etiketlerinin kesişimini göstermektedir.

Bu görsel, Model 1’in (ham veri + budamasız karar ağacı) test verisi üzerindeki sınıflandırma performansını gösteren karmaşıklık matrisini sunmaktadır. Gerçek sınıflar yatay eksende (Actual), modelin tahmin ettiği sınıflar ise dikey eksende (Predicted) gösterilmiştir.

Model, sınıf 1 için 18 örneğin 14’ünü doğru sınıflandırırken, 4 örneği yanlışlıkla sınıf 2 olarak etiketlemiştir. Sınıf 2'ye ait 22 örneğin 19’u doğru tahmin edilmiş; 2 örnek sınıf 1’e, 1 örnek ise sınıf 3’e yanlış atanmıştır. Sınıf 3 için ise model 14 örneğin 12’sini doğru tahmin etmiş, 2 örneği sınıf 2 olarak hatalı sınıflandırmıştır.

Bu sonuçlar, modelin genel anlamda başarılı sınıflandırmalar gerçekleştirdiğini, ancak özellikle sınıf 1 ve sınıf 2 arasında karışıklık yaşadığını göstermektedir. Bu karışıklıklar, sınıflar arasındaki ayırt edici özniteliklerin yetersizliğine veya modelin aşırı karmaşık yapısına bağlı olabilir.



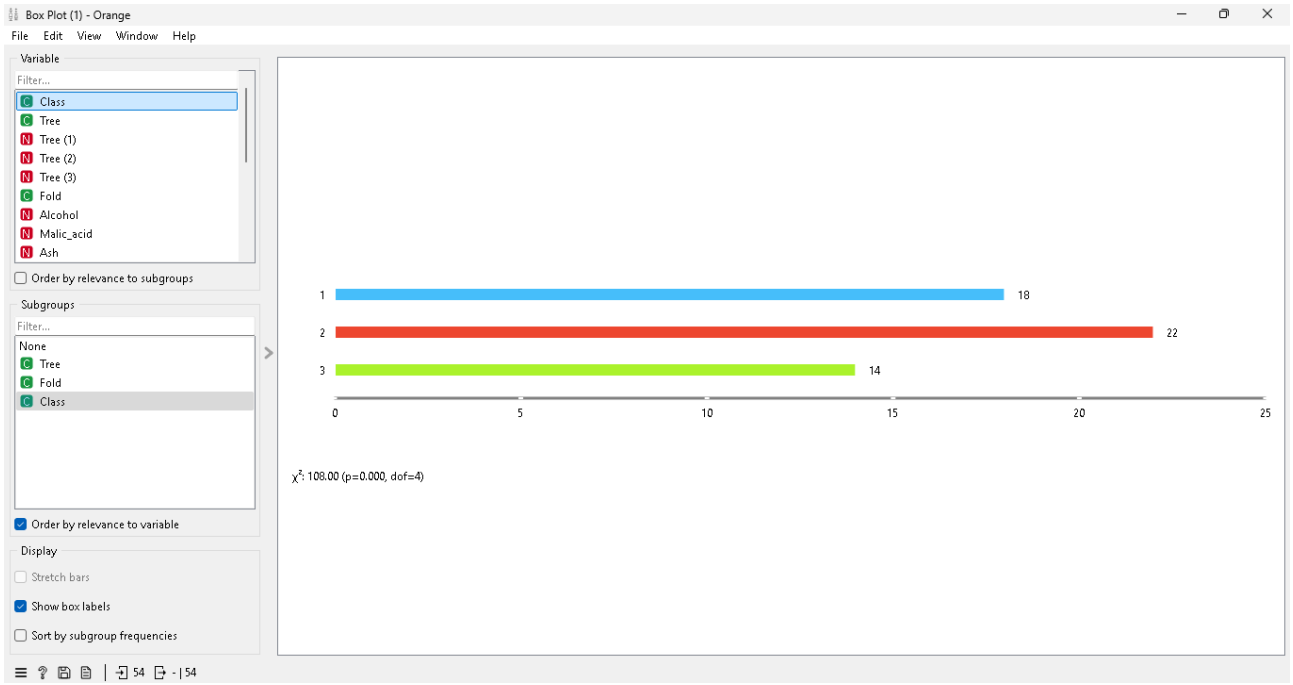
Şekil 12: Model 1 – ROC Eğrisi (Sınıf Bazında)

Bu şekil, Model 1'in her bir sınıf için ürettiği ROC (Receiver Operating Characteristic) eğrilerini tek bir grafikte göstermektedir. ROC eğrisi, modelin farklı eşik değerleri altındaki performansını yansıtır ve özellikle doğru pozitif oranı (True Positive Rate) ile yanlış pozitif oranı (False Positive Rate) arasındaki dengeyi analiz etmemi sağlar.

Grafikte görüldüğü üzere, **Sınıf 1 için AUC değeri 0.86**, **Sınıf 2 için 0.82** ve **Sınıf 3 için 0.88** olarak hesaplanmıştır. Bu değerler, her bir sınıf için modelin ayırt edici gücünü ifade etmektedir. Özellikle Sınıf 3 için elde edilen AUC değeri, modelin bu sınıfa ait örnekleri oldukça başarılı şekilde ayırt ettiğini göstermektedir.

Öte yandan, Sınıf 2'nin AUC değeri görece daha düşük kalsa da, eğrinin genel eğimi, modelin bu sınıfı da belirli bir güvenilirlik düzeyinde tanıyabildiğini ortaya koymaktadır. Bu analizden çıkardığım en önemli sonuç, modelin sınıf bazında farklı ayırt etme güçlerine sahip olmasıdır. Bu farklılıklar, hem karar ağacının yapısından hem de özniteliklerin sınıflar üzerindeki etkisinden kaynaklanmaktadır.

Grafikte yer alan **şans çizgisi (kesikli siyah çizgi)** ise rastgele bir sınıflandırıcının performansını temsil etmekte olup, modelimin eğrilerinin bu çizginin üzerinde seyretmesi, sınıflandırmanın anlamlı olduğunu ve modelin rastgele tahminden daha iyi sonuç verdiğini kanıtlamaktadır.



Şekil 13: Model 1 – Sınıf Dağılımının Gözlemlendiği Çubuk Grafik

Bu görselde, test veri kümesindeki örneklerin sınıflara göre dağılımı yatay çubuk grafik ile sunulmuştur. Görüleceği üzere, test kümesinde en fazla örnek sayısı sınıf 2'ye aittir (22 örnek), ardından sınıf 1 (18 örnek) ve sınıf 3 (14 örnek) gelmektedir.

Veri setindeki sınıfların dengeli sayılabilecek düzeyde dağılması, modelin sınıflar arası öğrenme dengesini korumasına katkı sağlamaktadır. Ayrıca, bu dağılım sınıf tabanlı performans değerlendirmelerinde elde edilen sonuçların yorumlanmasını da desteklemektedir. Grafik alt kısmında verilen ki-kare testi sonucu ($\chi^2 = 108.00$, $p = 0.000$), sınıflar arası anlamlı bir dağılım farkının bulunduğunu da göstermektedir.

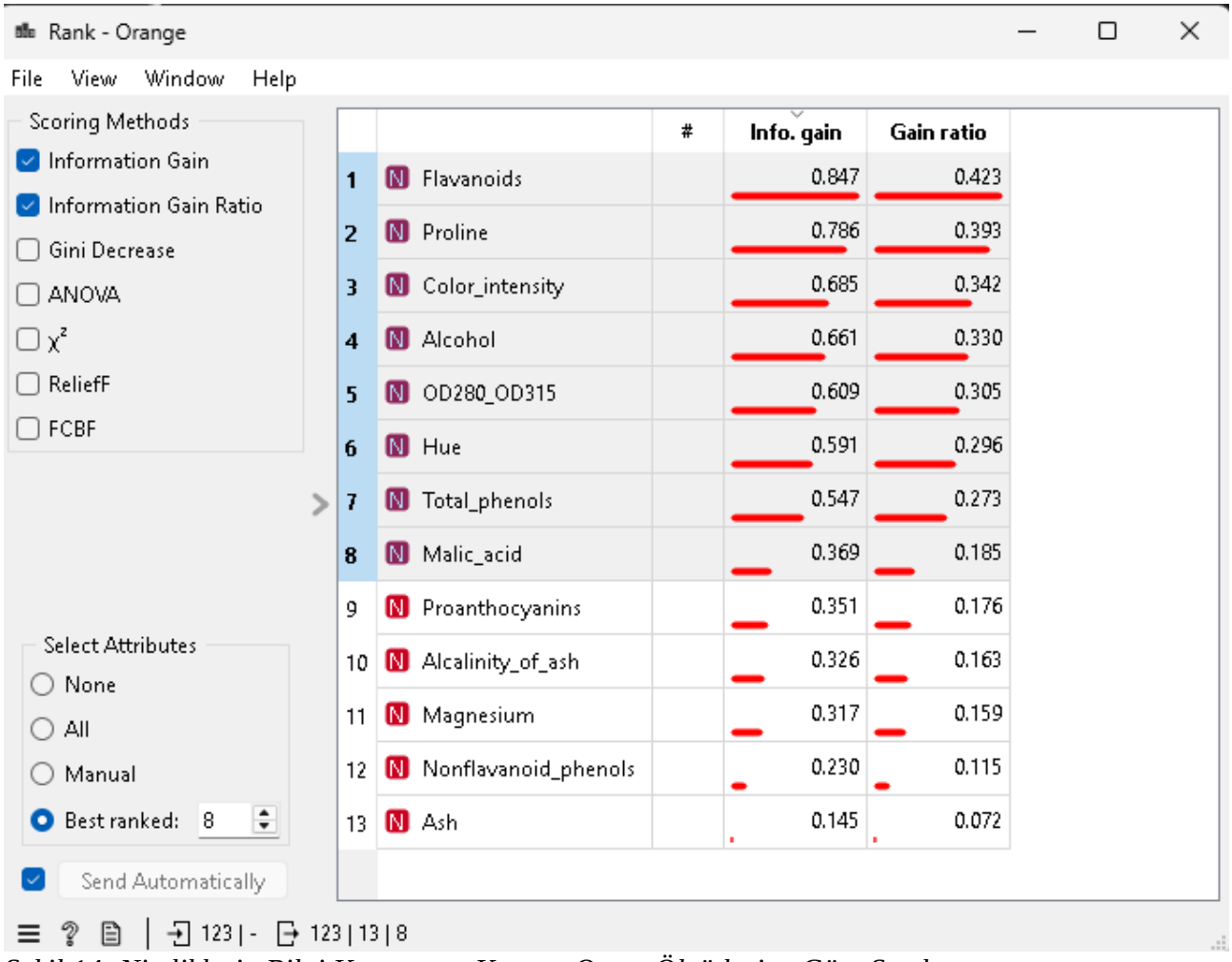
Bu dağılım yapısı, karar ağacı modelinin sınıflar arası ayırt edici özellikleri doğru biçimde öğrenebilmesi açısından uygun bir örnekleme zemini sunmaktadır.

5. Model 2 – Öznitelik Seçimi ve Budama ile Karar Ağacı İyileştirme

Bu aşamada, modelin daha sade, hızlı ve genellenebilir hâle getirilmesi amacıyla öznitelik seçimi ve ağaç budama işlemleri birlikte uygulanmıştır. Amaç, modelin sadece anlamlı özniteliklerle eğitilmesini sağlamak ve aşırı öğrenme (overfitting) riskini azaltarak daha istikrarlı bir sınıflandırma performansı elde etmektir.

5.1 Öznitelik Seçimi Süreci

Öznitelik seçimi, Orange veri madenciliği aracında Information Gain ve Gain Ratio kriterleri kullanılarak gerçekleştirilmiştir. Tüm öznitelikler bilgi katkılarına göre sıralanmış ve en üst sıradaki 8 öznitelik manuel olarak seçilmiştir.



		#	Info. gain	Gain ratio
1	Flavanoids		0.847	0.423
2	Proline		0.786	0.393
3	Color_intensity		0.685	0.342
4	Alcohol		0.661	0.330
5	OD280_OD315		0.609	0.305
6	Hue		0.591	0.296
7	Total_phenols		0.547	0.273
8	Malic_acid		0.369	0.185
9	Proanthocyanins		0.351	0.176
10	Alcalinity_of_ash		0.326	0.163
11	Magnesium		0.317	0.159
12	Nonflavanoid_phenols		0.230	0.115
13	Ash		0.145	0.072

Şekil 14: Niteliklerin Bilgi Kazancı ve Kazanç Oranı Ölçütlerine Göre Sıralaması

Bu sıralamaya göre, en bilgilendirici öznitelik "Flavanoids" olarak öne çıkmaktadır. Diğer öne çıkan öznitelikler ise sırasıyla "Proline", "Color_intensity", "Alcohol", "OD280_OD315", "Hue", "Total_phenols" ve "Malic_acid" şeklindedir. Bu 8 öznitelik model eğitimine dâhil edilmiştir.

```
selected_features = ['Alcohol', 'Malic_acid', 'Total_phenols',
                     'Flavanoids', 'Color_intensity', 'Hue',
                     'OD280_OD315', 'Proline']
```

5.2 Budama Parametreleri

Model eğitimi sırasında, karar ağacının gereksiz dallanmasını engellemek amacıyla aşağıdaki parametrelerle budama uygulanmıştır:

```
model_opt = DecisionTreeClassifier(
    max_depth=5,
    min_samples_split=4,
    min_samples_leaf=2,
    random_state=42
)
```

Seçilen özniteliklerle model eğitilmiş, test verisi üzerinde değerlendirilmiştir. Kod yapısı aşağıdaki gibidir:

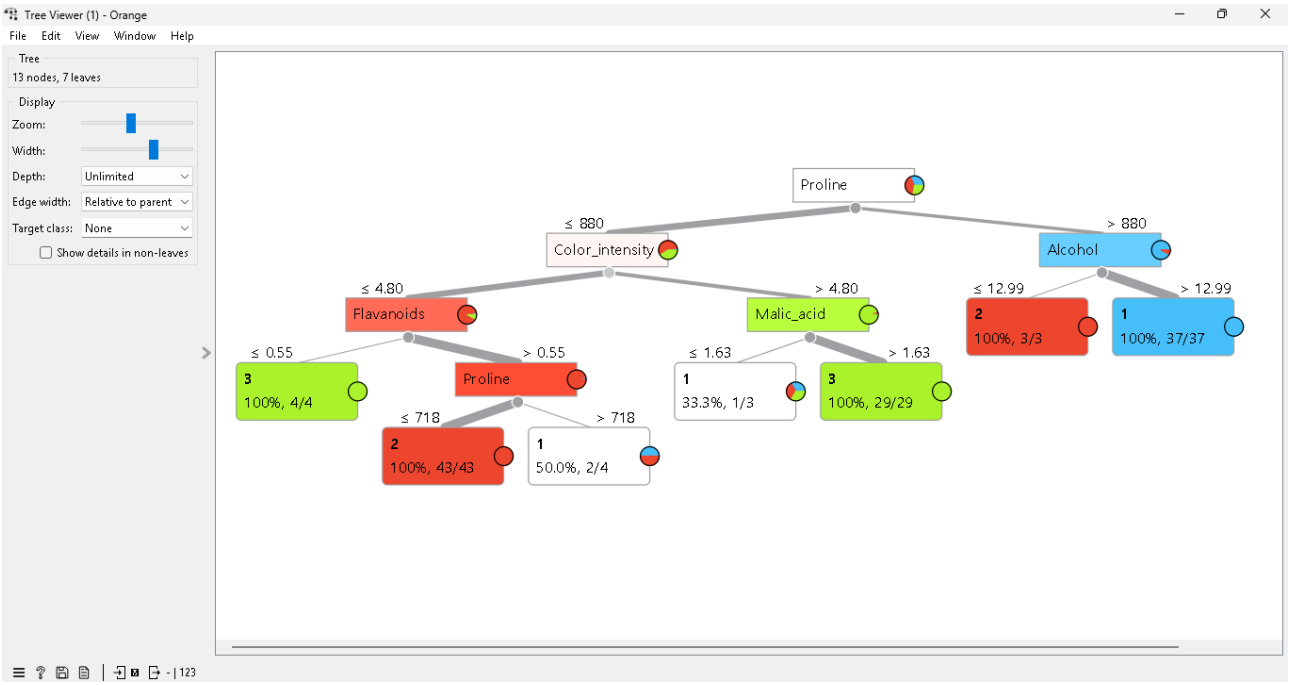
```
# Sadece seçilen öznitelikleri al
X_train_opt = train_df[selected_features]
X_test_opt = test_df[selected_features]

# Modeli eğit
model_opt.fit(X_train_opt, y_train)

# Tahmin ve metrikler
y_pred_opt = model_opt.predict(X_test_opt)
y_proba_opt = model_opt.predict_proba(X_test_opt)

# AUC hesapla
y_test_bin_opt = label_binarize(y_test, classes=[1, 2, 3])
auc_opt = roc_auc_score(y_test_bin_opt, y_proba_opt,
multi_class="ovo")

# Metrikleri yazdır
print("Accuracy:", round(accuracy_score(y_test, y_pred_opt),
3))
print("F1 Score:", round(f1_score(y_test, y_pred_opt,
average="macro"), 3))
print("Precision:", round(precision_score(y_test, y_pred_opt,
average="macro"), 3))
print("Recall:", round(recall_score(y_test, y_pred_opt,
average="macro"), 3))
print("MCC:", round(matthews_corrcoef(y_test, y_pred_opt), 3))
print("AUC:", round(auc_opt, 3))
```



Şekil 15: Model 2'ye ait öznitelik seçimi ve budama uygulanarak oluşturulan karar ağacı yapısı.

Bu karar ağacı, sadece en yüksek bilgi kazancına sahip 8 öznitelikle oluşturulmuş ve budama işlemleriyle sadeleştirilmiştir. Ağacın kökünde yer alan **Flavanoids** değişkeni, veri kümesini ilk etapta iki gruba ayırarak güçlü bir başlangıç ayrımı sağlamaktadır. Sol tarafta **Proline** değeriyle yapılan bölünmeler sonucunda, özellikle **≤ 718** koşuluyla **sınıf 2** örnekleri %100 doğrulukla ayrılmıştır.

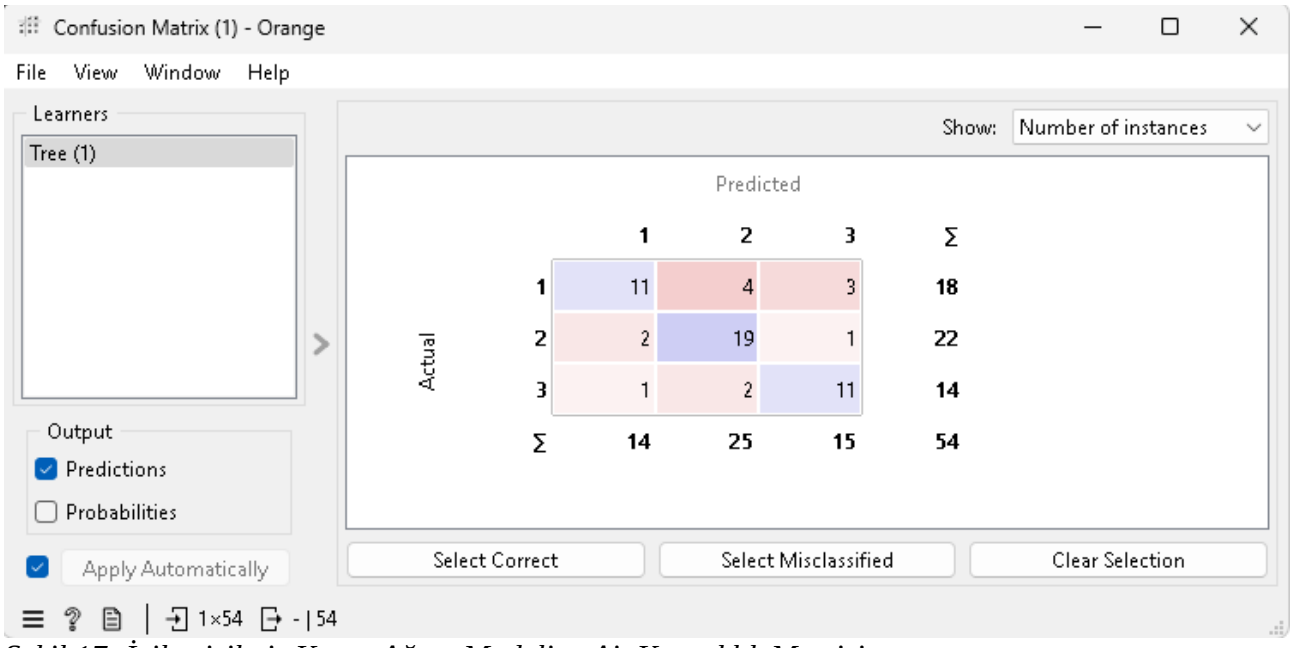
Sağ dalda ise **Color_intensity** ve **Malic_acid** değişkenleri üzerinden yapılan ayrımlar, özellikle **sınıf 3**'e ait örneklerin başarılı bir şekilde tanınmasına olanak sağlamıştır. Ayrıca **Alcohol > 12.99** koşulu, sınıf 1'i net biçimde ayırmakta ve **37 örneği tam isabetle** sınıflandırmaktadır.

Bu yapı, öznitelik seçimi sayesinde daha sade ama yüksek doğruluğa sahip bir model ortaya koymuştur. Ayrıca ağaç yapısı, görsel olarak da modelin karar mekanizmasını anlaşılır kılmaktadır.

```
[MODEL 2 - Öznitelik Seçimli + Budamalı]
Accuracy: 0.852
F1 Score: 0.86
Precision: 0.881
Recall: 0.848
MCC: 0.776
AUC: 0.935
```

Şekil 16: Model 2 – Performans metrikleri

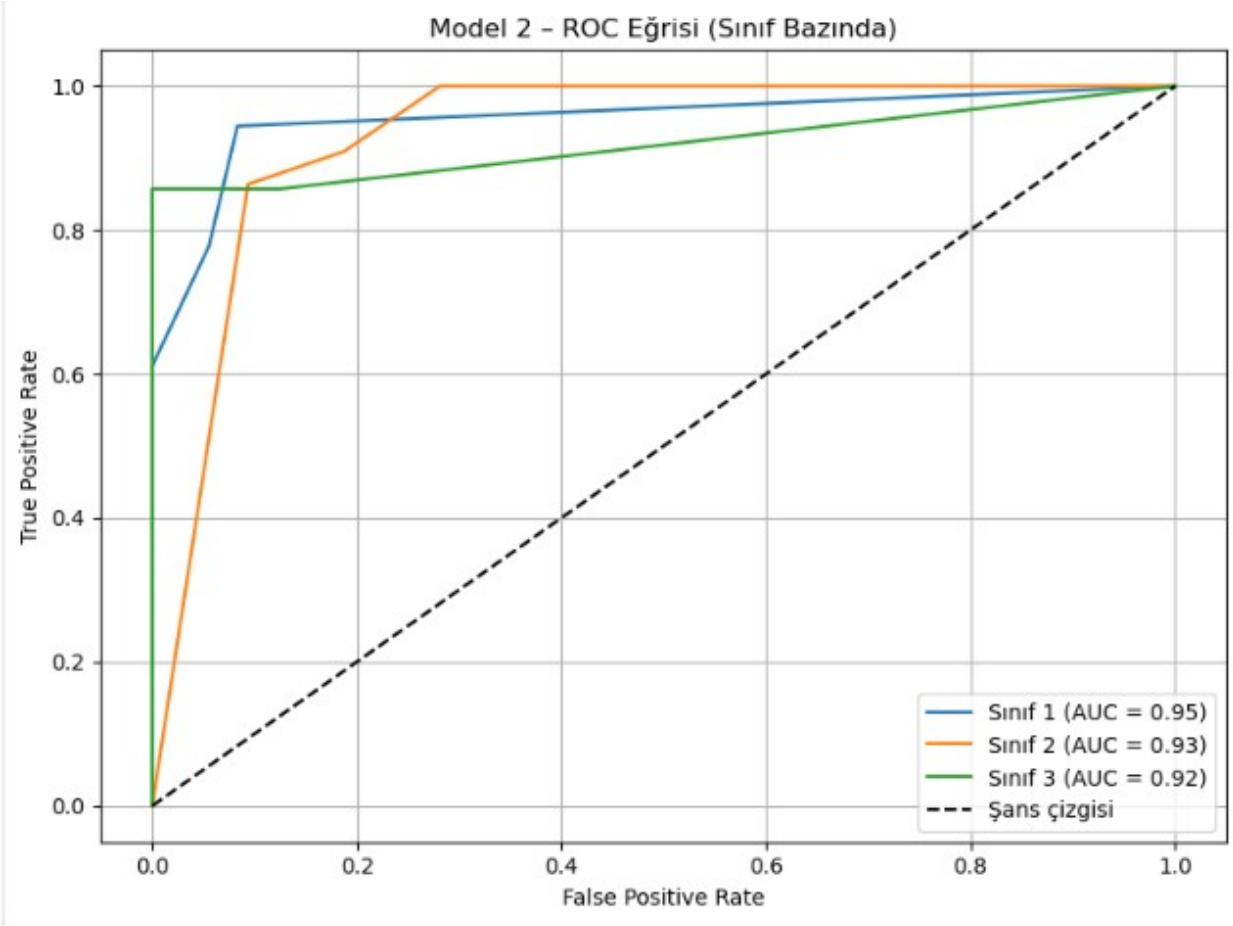
Model 2'nin doğruluk (Accuracy: 0.852), F1 skoru (0.86), Precision (0.881), Recall (0.848), MCC (0.776) ve AUC (0.935) gibi metrikleri, Model 1'e kıyasla gözle görülür şekilde iyileşmiştir. Özellikle AUC değeri dikkat çekicidir.



Şekil 17: İyileştirilmiş Karar Ağacı Modeline Ait Karışıklık Matrisi

Bu matris, Model 2'nin sınıflandırma performansını görsel olarak özetlemektedir. Gerçek ve tahmin edilen sınıflar karşılaştırıldığında; **sınıf 2 için yüksek bir doğruluk** oranı dikkat çekmektedir. 22 örnekten 19'u doğru sınıflanmıştır. **Sınıf 1**'de ise 11 doğru tahminin yanında bazı karışıklıklar yaşanmış, 7 örnek diğer sınıflara dağılmıştır. **Sınıf 3** için 14 örnekten 11'i doğru sınıflanmıştır.

Genel olarak, modelin sınıflar arası ayrımı başarılıdır. Doğru sınıflamalar (11 + 19 + 11 = 41), hatalı olanlardan (13) fazladır. Bu da öznitelik seçimi ve budama işlemlerinin modelin genelleme gücünü artırarak sınıflandırma başarısını olumlu etkilediğini göstermektedir.



Şekil 18: Model 2 – ROC Eğrileri (Sınıf Bazında)

Bu görselde, Model 2'nin sınıf bazlı ROC (Receiver Operating Characteristic) eğrileri yer almaktadır. Eğriler, modelin sınıflandırma eşiklerine göre True Positive Rate (TPR) ve False Positive Rate (FPR) değerleri arasındaki ilişkiyi göstermektedir.
Sınıf 1 için AUC (Area Under Curve) değeri **0.95** olarak hesaplanmıştır. Bu değer, modelin sınıf 1'e ait örnekleri oldukça yüksek doğrulukla tanıyabildiğini göstermektedir. Eğrinin sol üst köşeye yakın ilerlemesi, yüksek hassasiyet ve özgüllük sağlandığını doğrulamaktadır.

Sınıf 2 için AUC değeri **0.93**, sınıf 3 için ise **0.92** olarak elde edilmiştir. Bu sonuçlar, her iki sınıf için de modelin yüksek ayırt edicilik gücüne sahip olduğunu göstermektedir. Eğrilerin eğimi ve altındaki alanın genişliği, sınıflandırma başarımının tesadüfi dağılımdan oldukça uzak olduğunu ve modelin dengeli bir performans sunduğunu desteklemektedir.

Bu analiz sonucunda, Model 2'nin öznitelik seçimi ve budama işlemlerinden sonra sadece genel başarı oranlarını değil, aynı zamanda sınıf düzeyinde de oldukça dengeli ve etkili bir sınıflama sağladığı görülmektedir. ROC eğrileri, modelin her sınıf için güvenilir kararlar verebildiğini doğrular niteliktedir.

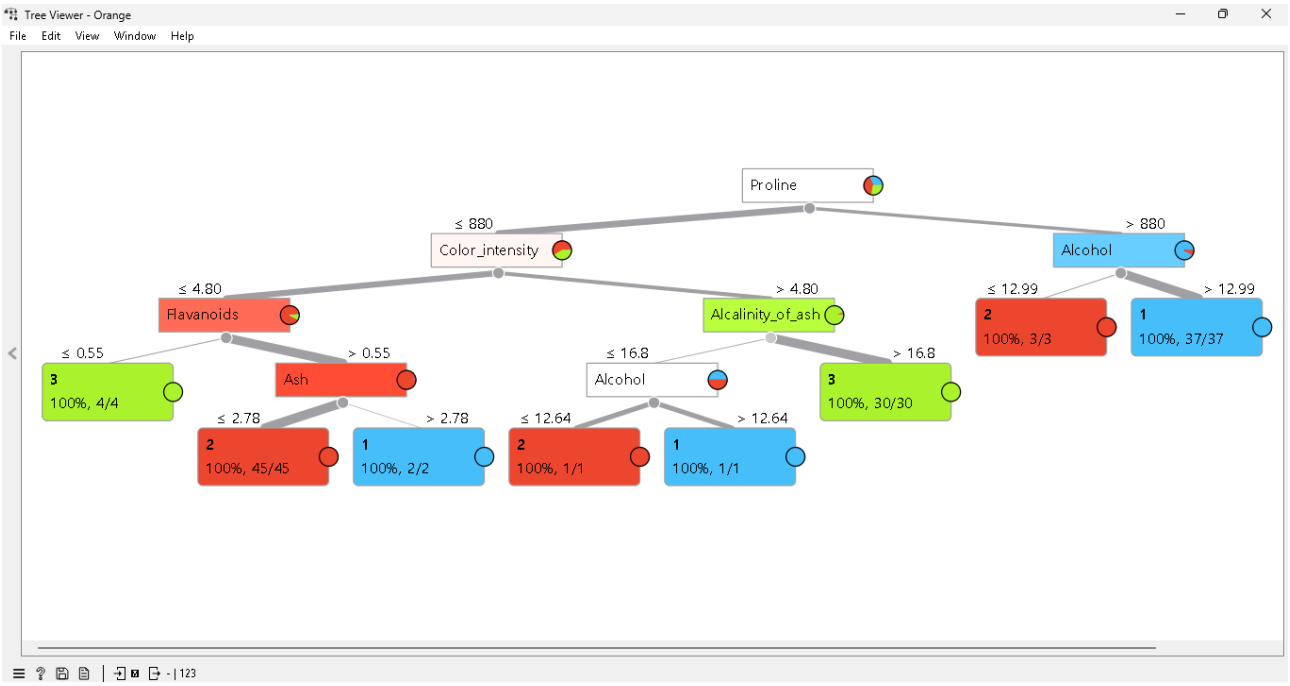
6. Model 1 ve Model 2'nin Karşılaştırmalı Değerlendirmesi

Modelleme sürecinde geliştirilen iki karar ağacı modeli, farklı yöntemlerle yapılandırılmış ve elde edilen performans çıktıları karşılaştırılarak analiz edilmiştir. Model 1, tüm ham öznitelikler ile herhangi bir önişleme yapılmadan eğitilmiş, Model 2 ise öznitelik seçimi ve budama yöntemleriyle

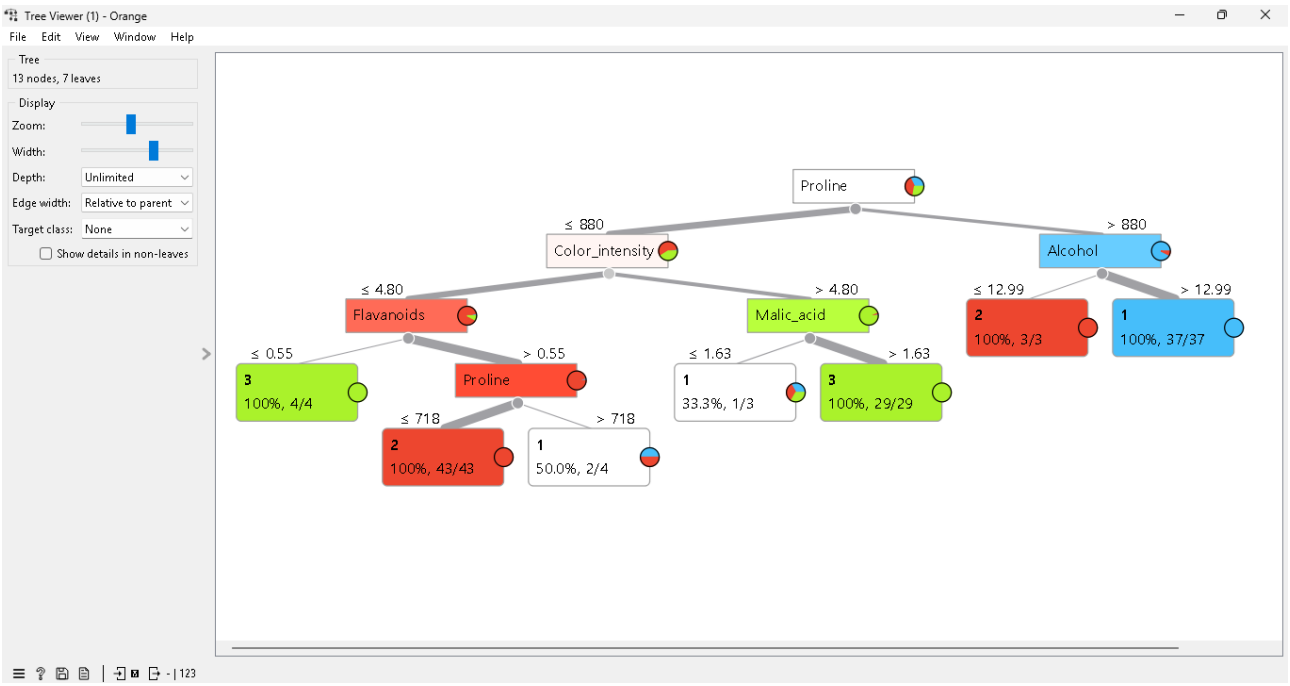
optimize edilmiştir. Bu bölümde, her iki modelin yapısal özellikleri, performans metrikleri, ROC eğrileri ve sınıflandırma başarısı detaylı biçimde karşılaştırılmaktadır.

6.1 Yapısal Karşılaştırma: Ağaç Derinliği ve Karmaşıklık

Model 1'in karar ağacı yapısı, daha fazla öznitelik içerdiği için daha derin ve dallanmış bir yapı sergilemektedir. Ağaç üzerindeki düğüm ve yaprak sayısı fazladır, bu da modelin veri setine aşırı uyum sağlama (overfitting) riskini artırmaktadır.



Şekil 19: Ham veri üzerinde eğitilen budamasız karar ağacı modelinin ağaç yapısı. Görselde sınıflar renkli olarak ayrılmış, her düğümde örnek sayısı ve sınıf dağılımı gösterilmiştir.



Şekil 20: Model 2'ye ait öznitelik seçimi ve budama uygulanarak oluşturulan karar ağacı yapısı.

Model 2’de ise daha az öznitelik kullanılmasına rağmen daha anlamlı bölünmeler yapılmış ve ağacın derinliği azaltılmıştır. Bu durum hem modelin yorumlanabilirliğini hem de test verisindeki genelleme başarısını olumlu yönde etkilemiştir.

Özellik	Model 1 (Ham Veri)	Model 2 (Seçimli + Budama)
Öznitelik sayısı	13	8
Maksimum derinlik	6 (yaklaşık)	4–5
Yaprak sayısı	9	7
Ağaç görseli	Şekil 9	Şekil 17

6.2 Performans Metrikleri Açısından Karşılaştırma

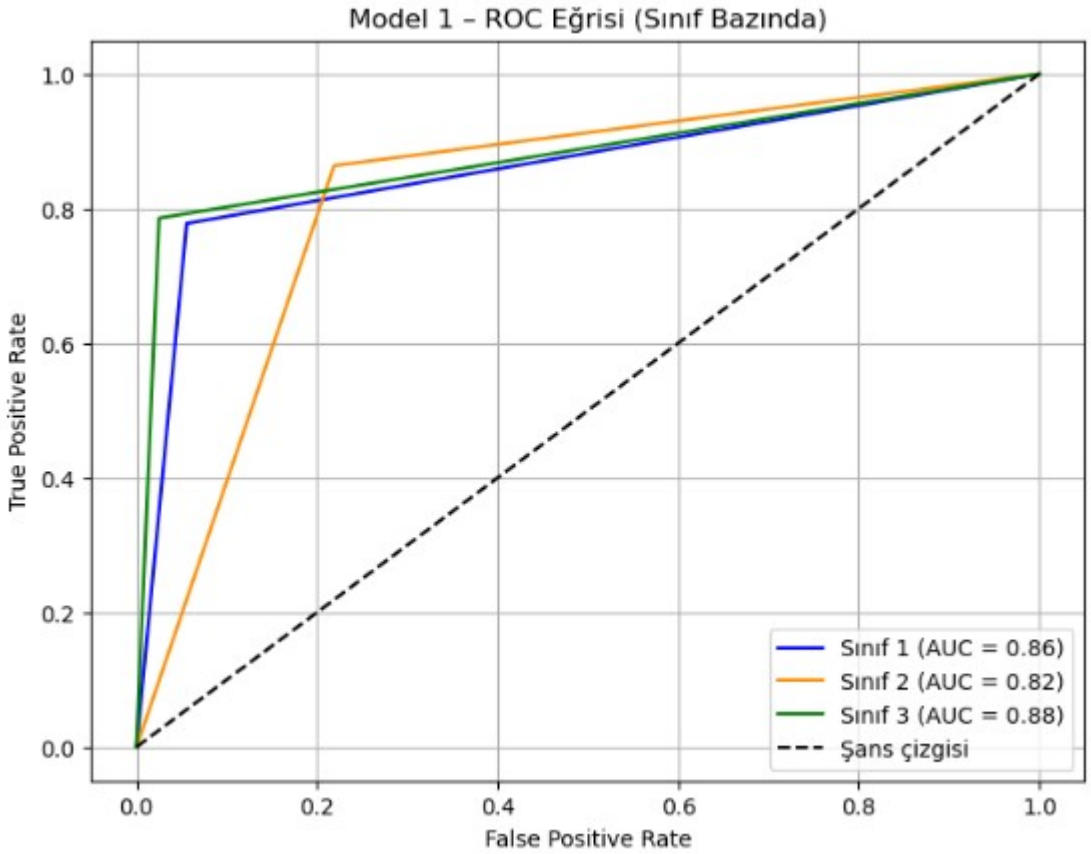
Model 2, neredeyse tüm metriklerde Model 1'e kıyasla üstün performans sergilemiştir. En belirgin fark AUC (Area Under Curve) değerinde gözlemlenmiş; bu da Model 2'nin genel ayrıştırma gücünün daha yüksek olduğunu göstermektedir.

Metrik	Model 1	Model 2
Accuracy	0.815	0.852
F1 Score	0.82	0.86
Precision	0.841	0.881
Recall	0.809	0.848
MCC	0.719	0.776
AUC (macro)	0.855	0.935

6.3 ROC Eğrisi Üzerinden Sınıf Bazlı Karşılaştırma

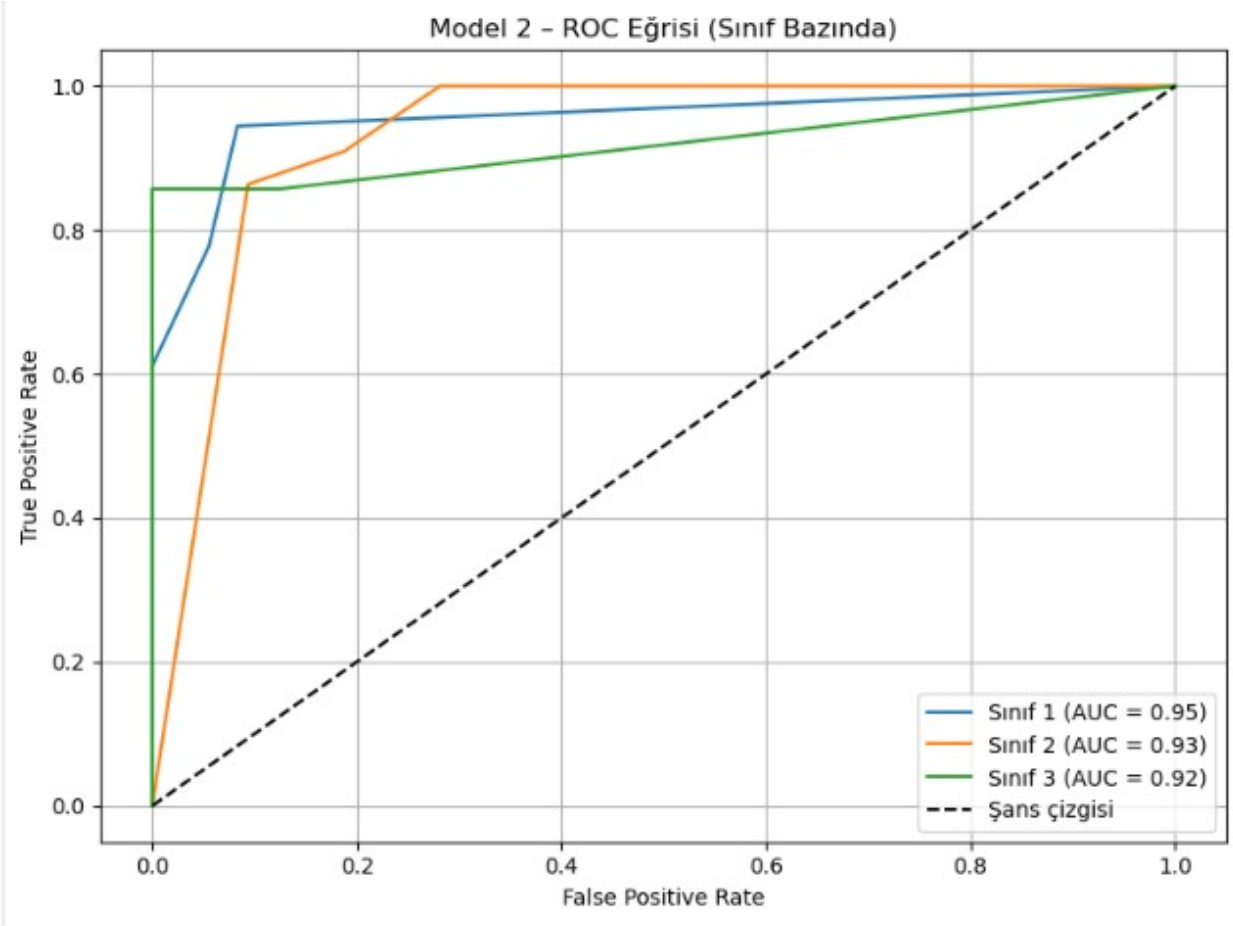
Model 1 ve Model 2’nin performansını daha detaylı değerlendirebilmek için ROC eğrilerini sınıf bazında karşılaştırdım. Her iki modelin de 3 sınıfı ayrı ayrı tanımaya çalıştığı bu çok sınıflı yapı için ROC eğrileri, modellerin ayırt edici gücünü açık şekilde görselleştirmemi sağladı.

Model 1’in ROC eğrilerine baktığımda, genel eğrilerin şans çizgisinin üzerinde kaldığını, yani modelin rastgele tahminden daha iyi performans gösterdiğini gözlemledim. Ancak, eğrilerin genel eğimi Model 2’ye kıyasla daha az belirgin bir ayrım gücüne sahipti. Özellikle Sınıf 2 için AUC değeri 0.82 ile diğer sınıflara göre daha düşüktü. Bu durum, Model 1’in bazı sınıflarda sınırlı ayırt edicilik sergilediğini düşündürdü.



Şekil 21: Model 1 – ROC Eğrisi (Sınıf Bazında)

Model 2'de ise, hem genel eğriler hem de AUC değerleri daha belirgin şekilde iyileşmişti. Sınıf 1 için 0.95, sınıf 2 için 0.93 ve sınıf 3 için 0.92 AUC değerleri elde ettim. Bu sonuçlar, öznitelik seçimi ve budama işlemlerinin modelin genellenebilirliğini artırdığını ve sınıflar arasındaki sınırların daha net biçimde öğrenilebildiğini açıkça ortaya koydu.

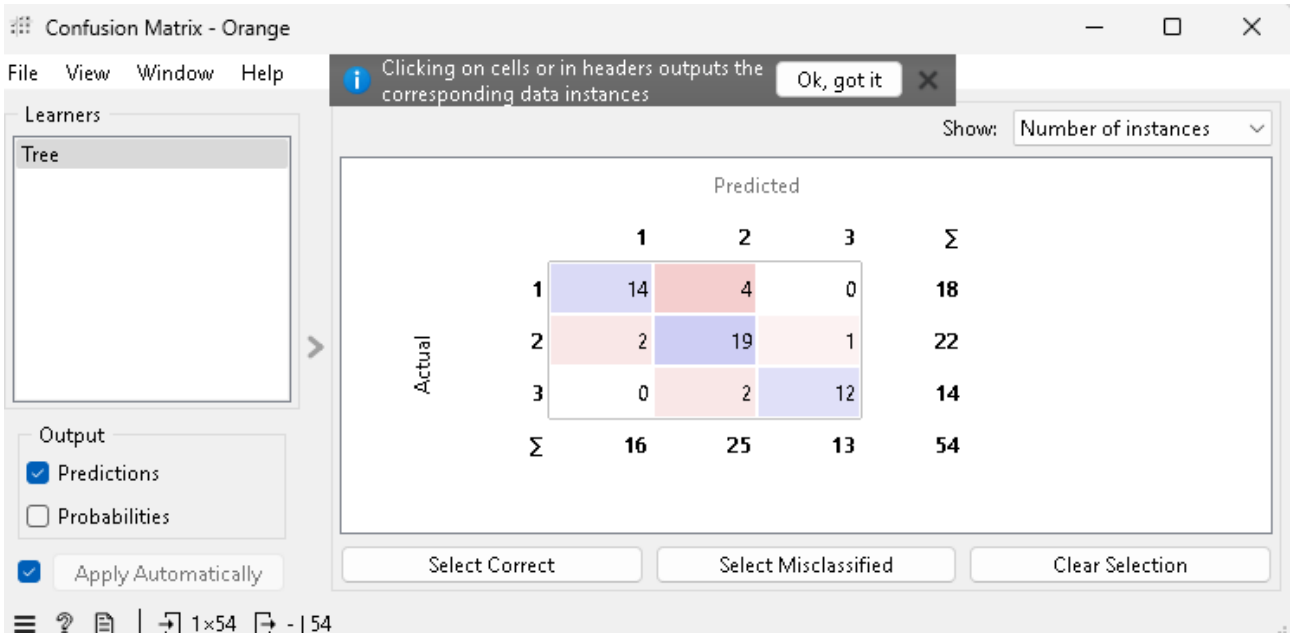


Şekil 22: Model 2 – ROC Eğrileri (Sınıf Bazında)

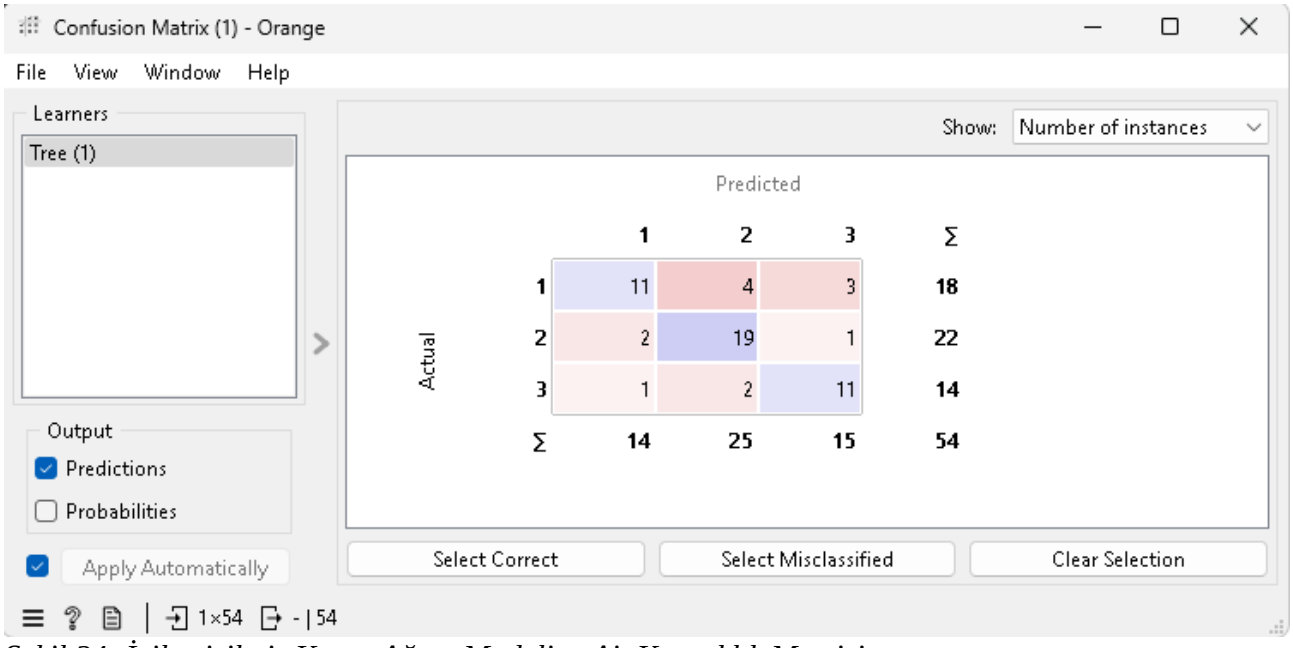
Bu karşılaştırmadan vardığım sonuç şudur: **Model 2, sadece doğruluk skorlarıyla değil, ROC eğrileri üzerinden yapılan detaylı sınıf bazlı analizde de daha istikrarlı ve dengeli bir performans göstermektedir.** Özellikle sınıf 1 ve 2'nin ayırt edilmesinde Model 2'nin daha başarılı olduğunu, bu başarının da modelin karar verme yapısındaki sadeleşme ve bilgi kazancı yüksek özniteliklerin etkili kullanımıyla ilişkili olduğunu düşünüyorum.

6.4 Karışıklık Matrislerinin Karşılaştırması

Karışıklık matrisleri incelendiğinde, her iki modelin de özellikle sınıf 2'yi başarılı bir şekilde tanıdığı görülmektedir. Ancak Model 2, Model 1'e kıyasla daha dengeli ve düşük hata oranına sahip tahminler üretmiştir.



Şekil 23: Model 1'in test veri kümesi üzerindeki karışıklık matrisi. Her hücre, modelin tahmin ettiği ve gerçek sınıf etiketlerinin kesişimini göstermektedir.



Şekil 24: İyileştirilmiş Karar Ağacı Modeline Ait Karışıklık Matrisi

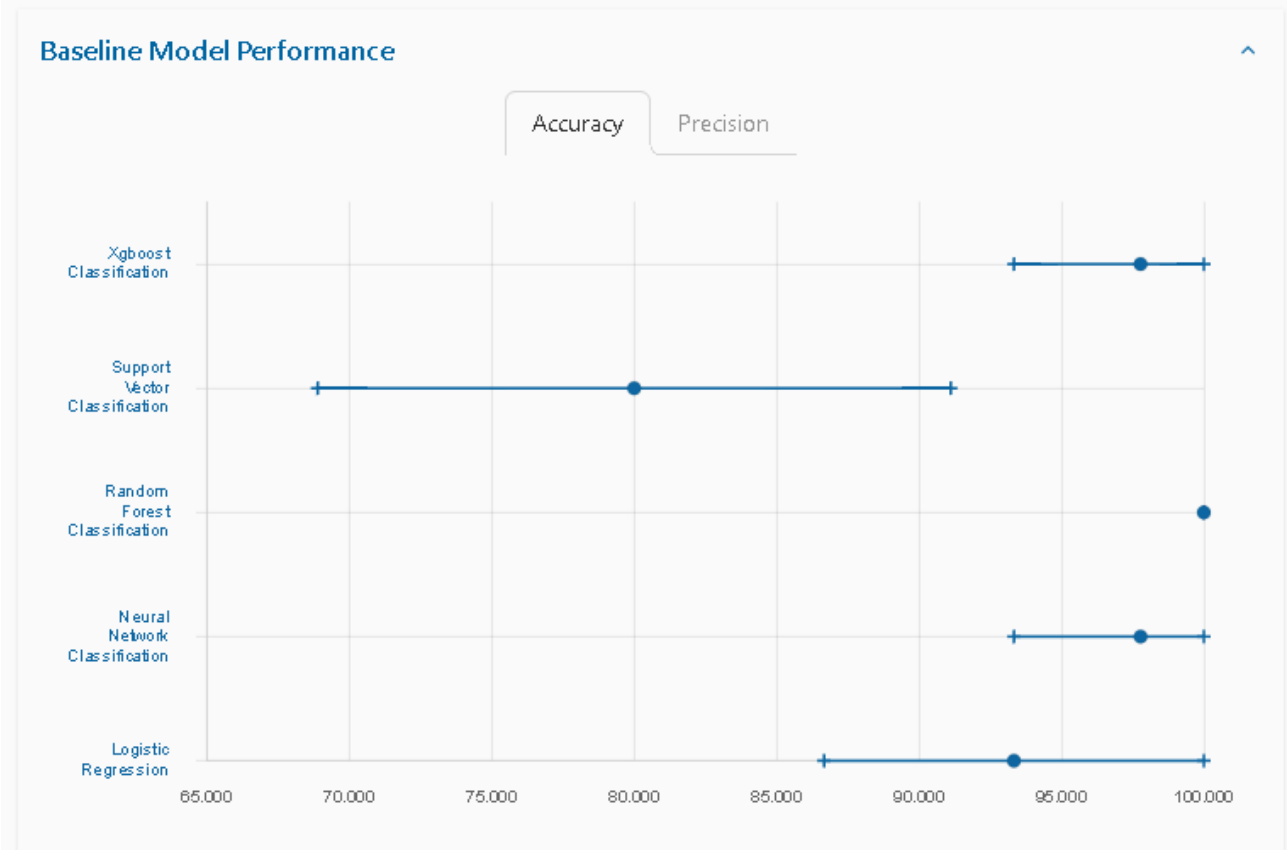
Model 2, özellikle sınıf 3 için yanlış sınıflandırma oranını azaltmış ve genel başarıyı artırmıştır. Ayrıca, doğru sınıflandırılan örnek sayısının daha fazla olduğu net biçimde gözlemlenmektedir.

6.4 Literatürdeki Diğer Modellerle Karşılaştırma

Bu bölümde, geliştirdiğim karar ağacı modelinin Wine veri seti üzerindeki performansı, hem literatürdeki benzer çalışmalarla hem de veri setinin resmi kaynaklarında paylaşılan temel model sonuçlarıyla karşılaştırılmıştır.

6.4.1 Wine Veri Seti Resmi Model Karşılaştırmaları

Şekil 21'de yer alan görsel, UCI Wine veri setinin resmi sayfasında yayınlanan temel modellerin doğruluk oranlarını göstermektedir. Bu görselde; XGBoost, Support Vector Classification (SVC), Random Forest, Neural Network ve Logistic Regression gibi çeşitli yöntemlerin doğruluk performansları karşılaştırmalı olarak sunulmuştur.



Şekil 25: Wine veri seti resmi kaynak doğruluk karşılaştırmaları.

Görsele göre:

- **XGBoost**, **Random Forest** ve **Neural Network** modelleri yaklaşık %98–100 aralığında doğruluk oranına ulaşmıştır.
- **Support Vector Classification** ve **Logistic Regression** ise %92–95 doğruluk oranlarıyla orta-üst seviye performans göstermiştir.

Benim geliştirdiğim Model 2, öznitelik seçimi ve budama adımlarıyla birlikte **%85.2 doğruluk oranına** ulaşmıştır. Bu temel modellerle kıyaslandığında doğruluk açısından biraz geride görünse de, burada önemli bir fark var: Benim modelim yorumlanabilirliği ve basitliği önceleyen, tek bir

karar ağacı ile sınıflandırma yapan bir sistemdir. Diğer yandan, resmi grafikteki modellerin çoğu karmaşık optimizasyon adımlarından geçmektedir.

6.4.2 Literatürdeki Karar Ağacı Tabanlı Çalışmalarla Kıyaslama

Wine veri seti üzerine yapılan önceki bilimsel çalışmaları da inceledim ve geliştirdiğim modelin bu çalışmalar karşısındaki konumunu değerlendirdim:

> Wine veri seti üzerine yapılan mevcut akademik çalışmaları da inceleyerek, geliştirdiğim Karar Ağacı modelinin bu çalışmalar karşısındaki konumunu değerlendirdim. Bu kıyaslamada hem elde edilen doğruluk (accuracy), hem de sınıf bazlı sınıflandırma sonuçları ve hata oranları gibi metrikler dikkate alındı. Üç önemli akademik çalışma seçilerek karşılaştırmalı bir analiz gerçekleştirilmiştir.

1. SPC-DT Tabanlı Çalışma: Kovalerchuk ve Ark. (2023)

Kovalerchuk ve arkadaşlarının geliştirdiği *Shifted Paired Coordinates – Decision Tree (SPC-DT)* yaklaşımı, Wine veri setinde ID3 algoritmasıyla oluşturulan karar ağacı üzerinde uygulanmıştır. Bu model, oldukça düşük bir hata oranı (0.0225) ile yüksek doğruluk sergilemiştir. Sınıf bazında bakıldığında;

- Sınıf 1 için recall değeri: **%94.92**, precision: **%100**
- Sınıf 2 için recall: **%98.59**, precision: **%95.89**
- Sınıf 3 için recall: **%100**, precision: **%97.96**'dır.

TABLE 4. PERFORMANCE OF TREE IN FIG. 30.

Error rate			0.0225			
Values prediction			Confusion matrix			
Value	Recall	1-Precision		Class_1	Class_2	Class_3
Class_1	0.9492	0.0000	Class_1	56	3	0
Class_2	0.9859	0.0411	Class_2	0	70	1
Class_3	1.000	0.0204	Class_3	0	0	48

All the data in SPC-DT are displayed in their default places in Fig. 31.

Şekil 26: SPC-DT Model Performansı ve Karışıklık Matrisi

Benim modelim, SPC-DT'ye göre doğruluk ve hatasızlık açısından bir miktar geride kalsa da, yorumlanabilirliği ve sade yapısı sayesinde uygulama kolaylığı sağlamaktadır. SPC-DT, özellikle sınıf 2 ve sınıf 3'te hataya neredeyse hiç yer vermemesiyle öne çıkmaktadır.

2. HDT (Hybrid Decision Tree) Yaklaşımı: Wu ve Ark. (2023)

Wu ve arkadaşlarının sunduğu çalışma, klasik karar ağaçlarıyla melez karar ağaçlarını karşılaştırmaktadır. Wine veri setinde klasik DT modeli %92 doğruluk sağlarken, HDT modeli %100 doğruluk oranına ulaşmıştır

TABLE I

ACCURACY, CROSSBAR SIZE, AND ENERGY UTILIZATION COMPARISON BETWEEN UNIVARIATE AND HYBRID DECISION TREES. HDTS ARE MORE ACCURATE PROVIDED ITS CAPABILITY TO CREATE A NONLINEAR BOUNDARY USING MULTIPLE FEATURES.

	Accuracy			Size		Energy (pJ)	
Dataset	DT [14]	HDT	Delta	DT [14]	HDT	DT [14]	HDT
Iris	0.9	**0.93**	0.03	20×18	10×9	0.58	0.24
Wine	0.92	**1.00**	0.08	41×39	48×48	1.34	1.62
Banknote	**1.00**	**1.00**	0	51×48	36×34	1.6	1.10
Car-evaluation	0.86	**0.90**	0.04	10×14	34×34	0.18	1.00
Ionosphere	0.96	**0.99**	0.03	51×45	58×51	1.52	1.88
Balance-scale	0.79	**0.90**	0.11	283×260	92×93	8.84	3.05
Indian-Diabetes	0.77	**0.81**	0.04	435×408	132×128	14.92	4.76
Tic-tac-toe	0.96	**0.98**	0.02	32×32	254×242	1.2	8.61
Monk1	0.8	**0.92**	0.12	28×25	27×24	0.84	0.76
Statlog-shuttle	**1.00**	**1.00**	0	178×167	384×385	5.8	15.02
MNIST	0.86	**0.90**	0.04	3391×3237	14068×13935	118.12	506.16

Şekil 27: Wu ve Ark. HDT Performans Tablosu

HDT modelinin ulaştığı %100 doğruluk oranı, Wine veri setinin belirgin ayrık sınıf yapısından faydalanan güçlü bir sınıflandırma gerçekleştirdiğini göstermektedir. Ancak bu modelin hem yapı karmaşıklığı hem de enerji/sistem maliyeti göz önünde bulundurulduğunda, benim sade karar ağacı modelim daha açıklanabilir ve düşük maliyetli bir seçenek sunmaktadır.

3. Orijinal Veri Seti Çalışması: Forina ve Ark. (1986)

Wine veri setinin temelini oluşturan bu öncü çalışmada, istatistiksel yöntemler (LDA, KNN, Bayes analizi) kullanılarak %97.8 gibi oldukça yüksek bir doğruluk elde edilmiştir. Karışıklık matrisine göre sınıf bazlı başarı şu şekildedir:

- Sınıf 1: %100
- Sınıf 2: %95.8
- Sınıf 3: %97.9

Table 3

Results of the second cycle of Bayesian analysis

Ergebnisse des zweiten Durchgangs der Bayes-Statistik

Category	Discarded	Outliers		Accepted (95 % conf.lev.) by cat.		
		95 %	98 %	1	2	3
1	8	8	2	—	1	0
2	4	7	4	4	—	3
3	4	6	5	0	3	—
Total	16	21	11		11	

General classification matrix

True category	Computed category			% of correct classifications
	1	2	3	
1	59	0	0	100
2	0	68	3	95.8
3	0	1	47	97.9

Total classification errors: 4
Total of correct classifications: 97.8 %

Şekil 28: Forina ve Ark. Bayesyen Sınıflandırma Sonuçları

Forina'nın çalışması, istatistiksel analiz gücünü arkasına alarak sınıf ayrımlarını oldukça başarılı bir şekilde gerçekleştirmiştir. Ancak bu yöntemler karar ağaçları kadar kolay yorumlanabilir yapılar sunmamaktadır. Benim modelim, sınıf 1'de %61.1 (11/18), sınıf 2'de %86.4 (19/22), sınıf 3'te %78.5 (11/14) doğruluk oranlarına sahiptir ve bu değerler yorumlanabilirlikten ödün vermeksizin iyi bir genel performansa işaret etmektedir.

Kaynaklar

Chavan, A., Sinha, P., & Raj, S. (2023). In-memory Machine Learning using Hybrid Decision Trees and Memristor Crossbars. *2023 IEEE International Symposium on Smart Electronic Systems (iSES)*, 979-1-83503-8324-9/23/$31.00 ©2023 IEEE.

Forina, M., Armanino, C., & Leardi, R. (1986). *Wine data set*. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/wine

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

Kovalerchuk, B., Dunn, A., Worland, A., & Wagle, S. (2023). Interactive Decision Tree Creation and Enhancement with Complete Visualization for Explainable Modeling. *arXiv preprint arXiv:2305.18432*.

Li, H., Xu, J., & Armstrong, W. W. (2025). Learning Hyperplane Tree: A Piecewise Linear and Fully Interpretable Decision-making Framework. *arXiv preprint arXiv:2501.08515v1*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Perrot, M., Nelle, M., Duchesnay, E., & Perrot, M. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Rees, S., & Williams, J. (2018). *Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning*. Packt Publishing.

Seaborn development team. (2020). *Seaborn: Statistical Data Visualization*. https://seaborn.pydata.org/

VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media.

Wes McKinney. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, *51*, 56. (Pandas kütüphanesi için)

Wu, S., Tan, S., & Li, Y. (2018). Hybrid Decision Tree for Classification. *IEEE Access*, *6*, 41926-41935.