



Hepatit Veri Setinin Karar Ağacı Algoritması ile Analizi

Muhammed Saltuk Yaşar – 18360859016

Bursa Teknik Üniversitesi





İşlenecek Konular

- Veri Seti
- Sınıflandırma Ve Karar Ağacı Algoritması
- Veri Ön İşleme
- Konfüzyon Matris
- Metrikler
- Analizler
- Diğer Çalışmalar
- Sonuç



Giriş:

- Hepatit veri setinde bulunan karar ağacı sınıflandırma algoritmasının performansını incelemektedir. Bu çalışma, medikal verilerin doğru bir şekilde sınıflandırılabilmesi için bir sınıflandırma modeli olarak karar ağacı kullanmanın etkinliğini vurgulamaktadır



Veri Seti:

- Bu veri seti, hepatit hastalığı ile ilgili klinik ve demografik verileri içeren bir tıbbi veri setidir. Toplamda 155 hasta verisi bulunmaktadır. Veriler, 19 farklı özellik (feature) içerir ve 2 farklı sınıf (class) bulunur.
- Bu veri seti, hastalık tanısı ve prognozu konusunda veri madenciliği çalışmaları için uygun bir veri kaynağıdır.
- Hepatit hastalığının etkili bir şekilde teşhis edilmesi ve tedavi edilmesi, hastaların hayat kalitesini artırabilir ve hayatlarını kurtarabilir
- Bu veri seti, UCI Machine Learning Repository'de bulunabilir ve tıp alanında veri madenciliği çalışmaları yapan araştırmacılar için önemli bir kaynak olabilir

Hepatit Veri Seti:

Attributes	Data Type	Values
Class (life)	Categorical	Ölü, Canlı
Age	Numerical	Numerical Values
Gender	Categorical	Male, Female
Steroid	Categorical	Yes, No
Antivirus	Categorical	Yes, No
Fatigue	Categorical	Yes, No
Malaise	Categorical	Yes, No
Anorexia	Categorical	Yes, No
Liver_Big	Categorical	Yes, No
Liver_Firm	Categorical	Yes, No
Spleen_Palpable	Categorical	Yes, No
Spiders	Categorical	Yes, No
Ascites	Categorical	Yes, No
Varices	Categorical	Yes, No
Bilirubin	Numerical	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
Alk_Phosphate	Numerical	33, 80, 120, 160, 200, 250
SGOT	Numerical	13, 100, 200, 300, 400, 500
Albumin	Numerical	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
Protime	Numerical	10, 20, 30, 40, 50, 60, 70, 80, 90
Histology	Categorical	Yes, No



Sınıflandırma

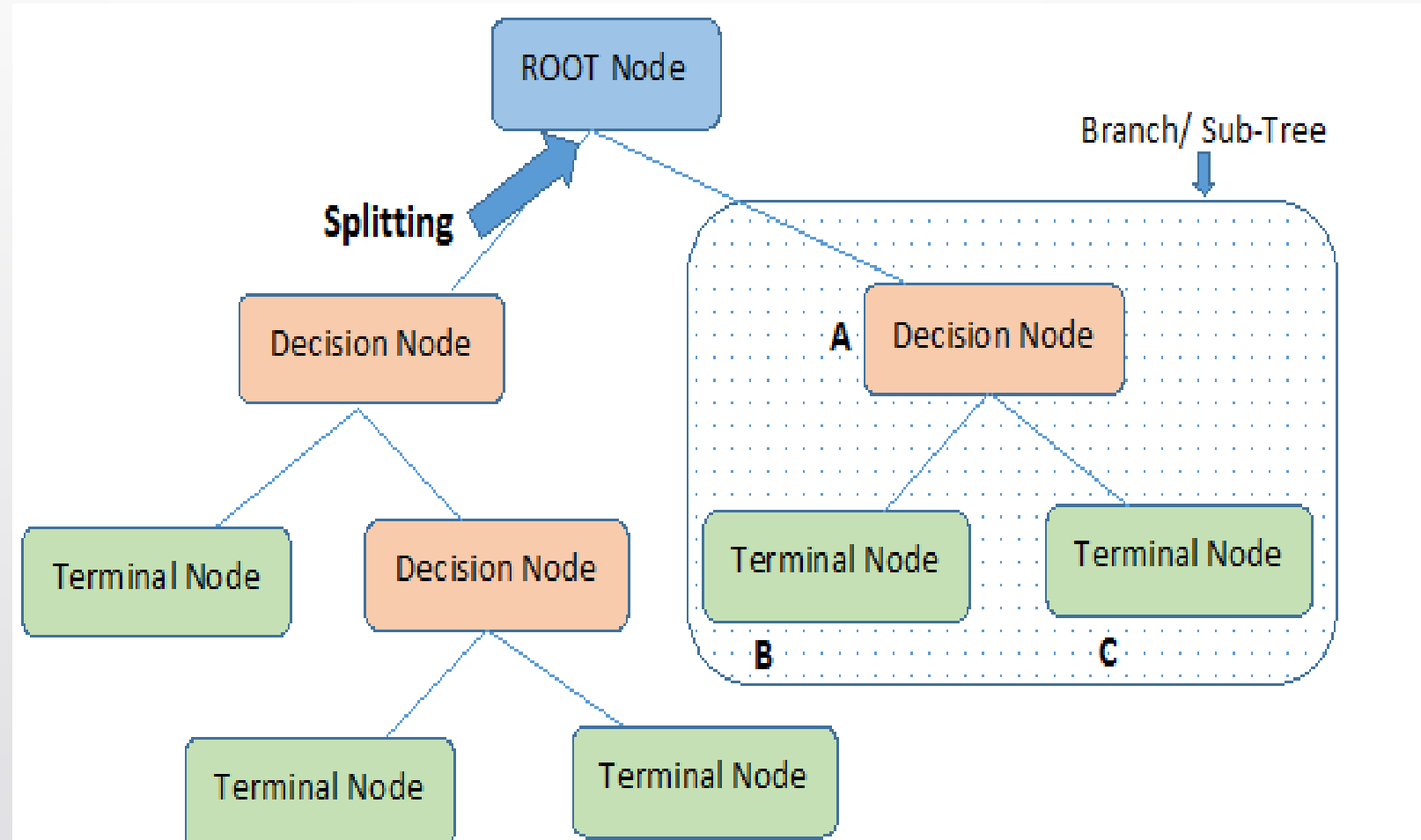
- Sınıflandırma algoritmaları, makine öğrenimi ve istatistik alanlarında kullanılan bir dizi teknikle, veri girişlerini belirli kategorilere göre sınıflandırır.
- Çok sayıda sınıflandırma tekniği ve algoritması mevcuttur. Bunlar arasında, en yaygın olarak kullanılan yöntemler arasında karar ağacı, destek vektör makineleri, k-NN, doğrusal ve lojistik regresyon, rastgele ormanlar ve yapay sinir ağları bulunmaktadır.



Karar Ağacı Algoritması

- Karar ağacı algoritmaları, bir veya daha fazla bağımsız değişkenin ölçümlerini kullanarak, kategorik bir bağımlı değişkenin sınıflarını tahmin etmek için kullanılan bir sınıflandırma yöntemidir.
- Karar ağacı algoritmaları, bir ağacın dallarını, gözlemleri iki veya daha fazla alt gruba bölen bir değişken ve eşik değeriyle oluşturur. Bu adım, gini indeksi, kazanç oranı, entropi ölçümleri gibi matematiksel algoritmalar kullanılarak gerçekleştirilir. Bu ağaç, tahmin doğruluğunu artırmak için gözlemleri yinelemeli olarak ayırır.

- Kök
- Düğüm
- Yaprak





Karar Ağacı Avantajları Ve Dezavantajları

- Kolay Anlaşılabilir
- **Esneklik:** Karar ağacı algoritmaları, verilerin farklı tiplerini işleyebilir. Sayısal, kategorik ve nominal verileri içeren verileri işleyebilirler
- **Çok amaçlılık:** Karar ağacı algoritmaları, hem sınıflandırma hem de regresyon problemlerini çözmek için kullanılabilir.
- **Overfitting:** Karar ağacı algoritmaları, eğitim verilerine aşırı uyum sağlayabilirler. Bu durum, test verilerinde düşük performansa neden olabilir.
- **Düşük Doğruluk:** Karar ağacı algoritmaları, diğer algoritmalarla kıyasla düşük doğruluk oranlarına sahip olabilirler.

Karar Ağaçlarında Dallanma

- Gini Endeksi (homojenlik)
- Entropy Ölçümü (düzensizlik)
- Hata tahmini

$$Gini = 1 - \sum_j p_j^2$$

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

$$E = 1 - \max(\hat{p}_{mk})$$



Veri Ön İşleme

- Veri ön işleme, makine öğrenimi modellerinin doğru ve etkili bir şekilde çalışabilmesi için veri setinin hazırlanmasıdır. Veri setlerindeki hatalar, aykırı değerler, eksik veriler, dengesiz sınıf dağılımı ve gereksiz özellikler, modelin performansını olumsuz yönde etkileyebilir. Bu nedenle, veri setleri öncelikle ön işlemeye tabi tutulur.

Veri Ön İşleme

```
# Veri kümesini yükle
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/hepatitis/hepatitis.data"
names = ['Class', 'Age', 'Sex', 'Steroid', 'Antivirals', 'Fatigue', 'Malaise', 'Anorexia', 'LiverBig',
data = pd.read_csv(url, names=names, na_values='?')

# Eksik verileri sil
data.dropna(inplace=True)

# Hedef değişkeni ayır
X = data.drop('Class', axis=1)
y = data['Class']

# Eğitim ve test verilerini ayır
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

- X değişkeni, sınıflandırma modelinde kullanılacak bağımsız değişkenleri (yani hedef değişkeni hariç diğer tüm değişkenleri) temsil eder ve y değişkeni, sınıflandırılacak hedef değişkeni olan 'Class' sütununu temsil eder.
- train_test_split işlevi, veri kümesini eğitim ve test setlerine ayırır. test_size argümanı, test verilerinin oranını belirler ve random_state argümanı, verilerin rastgele bölünmesini kontrol etmek için kullanılır.

Karar Ağacının Eğitilmesi

```
# Decision Tree modelini e it
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)

# Test verileriyle sınıflandırma yap
y_pred = clf.predict(X_test)

# Karar ağacı modelinin oluřturulması
decision_tree = DecisionTreeClassifier(random_state=42)
decision_tree.fit(X_train, y_train)
```

İlk olarak, `DecisionTreeClassifier()` sınıfından bir nesne oluřturulur ve bu nesne "clf" adlı deęiřkene atanır. Ardından, `fit()` y ntemi kullanılarak model, eęitim verileri  zerinde eęitilir. Daha sonra, `predict()` y ntemi kullanılarak, oluřturulan model test verileri  zerinde uygulanarak sınıflandırma yapılır ve "y_pred" adlı bir deęiřkene atanır. En son olarak, ikinci bir `DecisionTreeClassifier()` nesnesi oluřturulur ve bu nesne "decision_tree" adlı bir deęiřkene atanır. Bu nesne de aynı řekilde eęitim verileri  zerinde eęitilir.

Metrikler

Confusion Matrix	Predicted Class		
	Class	Class 1	Class 2
Actual Class	Class 1	True Positive (TP)	False Negative (FN)
	Class 2	False Positive (FP)	True Negative (TN)

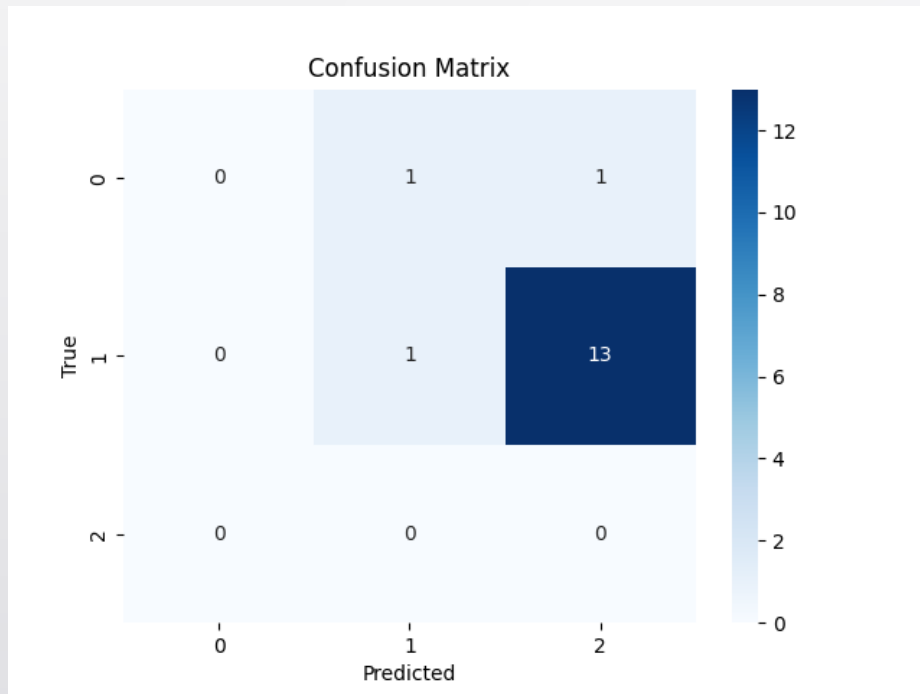
- Accuracy (doğruluk) bir sınıflandırma modelinin performansını ölçmek için kullanılan en temel ölçümlerden biridir. Modelin ne kadar doğru tahmin yaptığını gösterir.

Maliyete Duyarlı Metrikler

$$\text{Recall} = \text{TPR} = \frac{TP}{TP + FN}$$
$$\text{Precision} = \frac{TP}{TP + FP}$$
$$\text{FPR} = \frac{FP}{FP + TN}$$
$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

- Precision (kesinlik): Gerçek pozitif sınıflandırmaların tüm pozitif sınıflandırmalar içindeki oranını ifade eder. Yani, ne kadar pozitif tahminin gerçekten doğru olduğunu ölçer.
- Recall (duyarlılık): Gerçek pozitif sınıflandırmaların tüm pozitif örnekler içindeki oranını ifade eder. Yani, gerçek pozitif örneklerin ne kadarını doğru bir şekilde tespit ettiğimizi ölçer.
- F-measure (F-skor): Precision ve recall değerlerinin harmonik ortalamasını ifade eder. Bu ölçüm, hem precision hem de recall değerlerinin önemli olduğu durumlarda kullanışlıdır. Formülü: $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$
- Accuracy (doğruluk): Doğru sınıflandırılan örneklerin tüm örnekler içindeki oranını ifade eder. Yani, sınıflandırmanın ne kadar doğru yapıldığını ölçer.

Konfüzyon Matrisi



- Konfüzyon matrisi, sınıflandırma problemlerinde bir modelin performansını değerlendirmek için kullanılan bir metriktir. Matris, gerçek sınıfların ve tahmin edilen sınıfların kesişiminde bulunan 4 farklı örneği gösterir: True Positive (TP), False Positive (FP), True Negative (TN) ve False Negative (FN).
- Konfüzyon matrisi, bir modelin performansını ölçmek için birçok metrik üretir, örneğin doğruluk oranı (accuracy), hassasiyet (precision), duyarlılık (recall) ve F-measure.

Metriklerimizi hesaplayan kod

```
147 # Confusion Matrix'i hesapla
148 cm = confusion_matrix(y_test, y_pred)
149
150 # True Positive Rate, True Negative Rate,
151 # False Positive Rate, False Negative Rate hesapla
152 tn, fp, fn, tp = cm.ravel()
153 tpr = tp / (tp + fn)
154 tnr = tn / (tn + fp)
155 fpr = fp / (fp + tn)
156 fnr = fn / (fn + tp)
157
158 # Performans ölçümleri
159 acc = accuracy_score(y_test, y_pred)
160 prec = tp / (tp + fp)
161 rec = tp / (tp + fn)
162 f1 = (2*prec*rec)/(prec+rec)
163
```

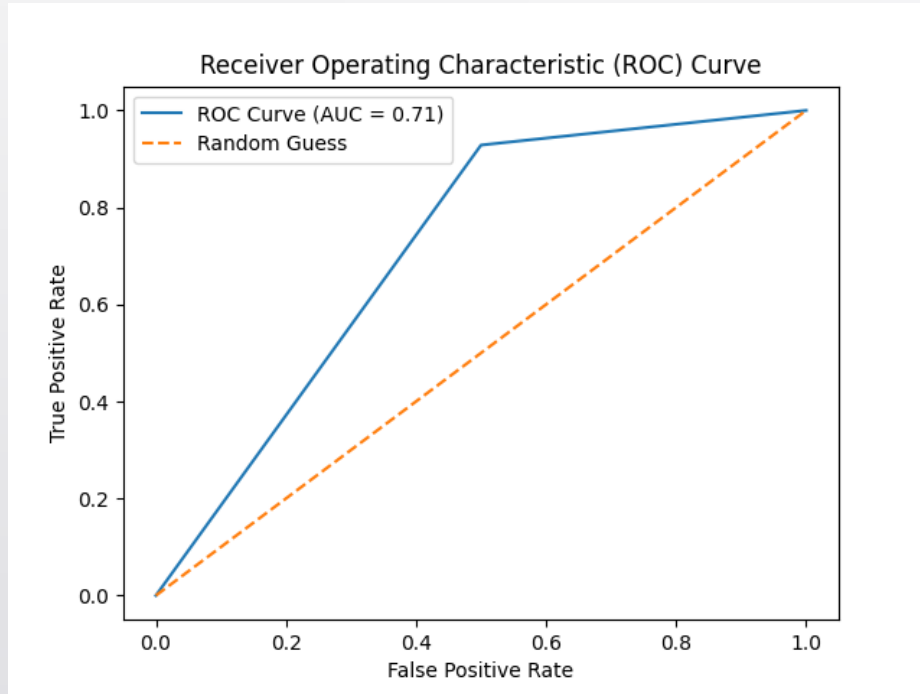
```
C:\Users\saltu\PycharmProjects\HepatitisDataSet
Accuracy: 0.875
Accuracy: 0.8750
Precision: 0.9286
Recall: 0.9286
F1 Score: 0.9286
True Positive Rate: 0.9286
True Negative Rate: 0.5000
False Positive Rate: 0.5000
False Negative Rate: 0.0714
```



Analiz

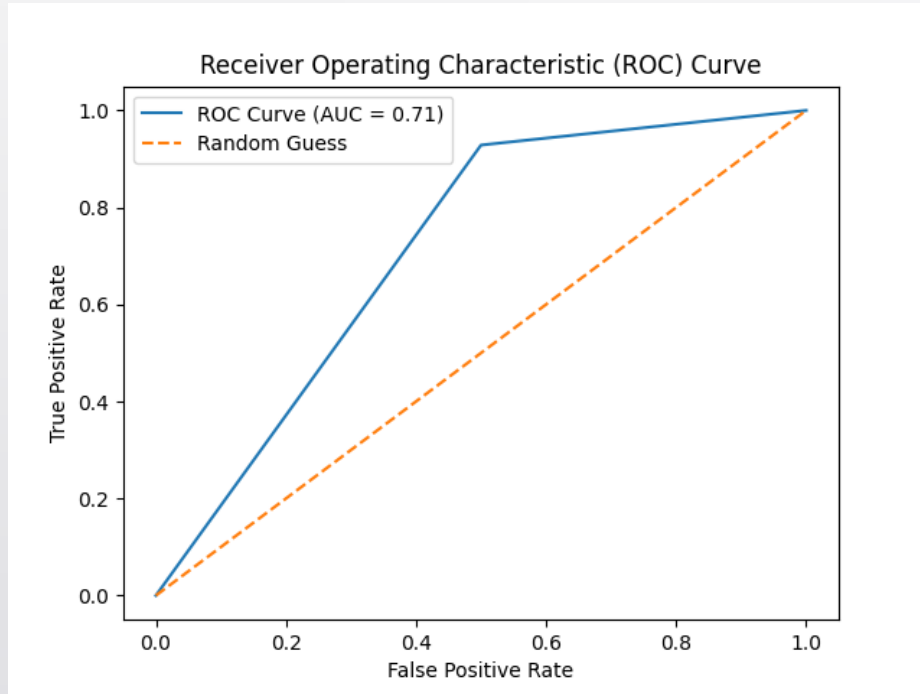
- Modelin doğruluğu %87,5 olarak ölçülmüştür, yani doğru sınıflandırma oranı oldukça yüksektir.
- Ayrıca, modelin pozitif tahminlerinin %92,86'sının gerçekten pozitif olduğu, gerçek pozitif örneklerin %92,86'sının doğru bir şekilde tespit edildiği ve F1 skoru %92,86 olduğu görülmüştür.
- Bu sonuçlar, modelin başarılı bir şekilde pozitif örnekleri belirleme ve gerçek pozitif örnekleri bulma konusunda iyi bir performans sergilediğini göstermektedir.

ROC Eğrisi (Curve)



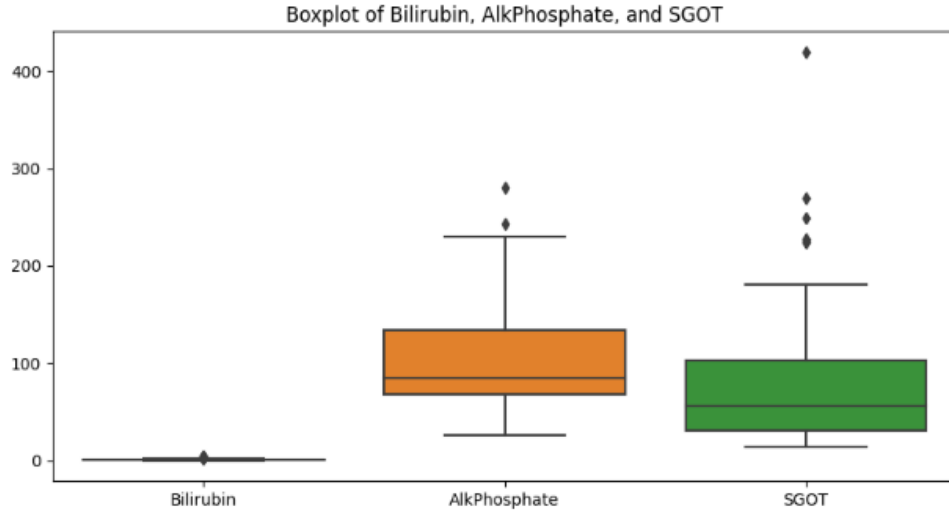
- ROC eğrisi, bir sınıflandırma modelinin duyarlılık (sensitivity) ve özgüllük (specificity, precision) performansını aynı anda gösterir.
- ROC eğrisinin altında kalan alan (AUC), bir sınıflandırma modelinin performansını ölçmek için kullanılan bir diğer önemli metriktir.
- AUC değeri, modelin rastgele sınıflandırmaya göre ne kadar iyi bir performans gösterdiğini ölçer. AUC değeri 1'e yaklaştıkça modelin performansı da o kadar yüksek kabul edilir.

ROC Eğrisi Analizi



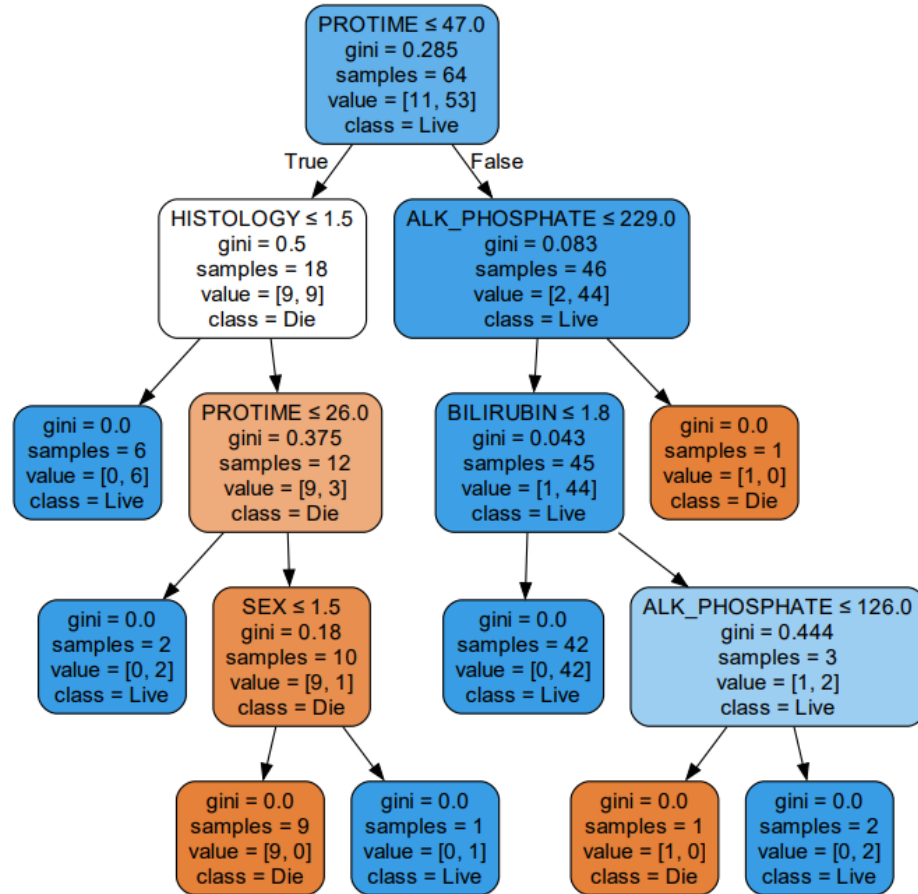
- Hepatit veri setinin ROC eğrisi, AUC değeri 0.71 olarak hesaplanmıştır.
- Bu, karar ağacı sınıflandırma modelinin iyi bir performans sergilediğini göstermektedir.
- Eğri, sol üst köşeye doğru yüksek bir eğim ile yükselmektedir, bu da modelin yüksek bir TPR'ye sahip olduğunu göstermektedir.

Boxplot



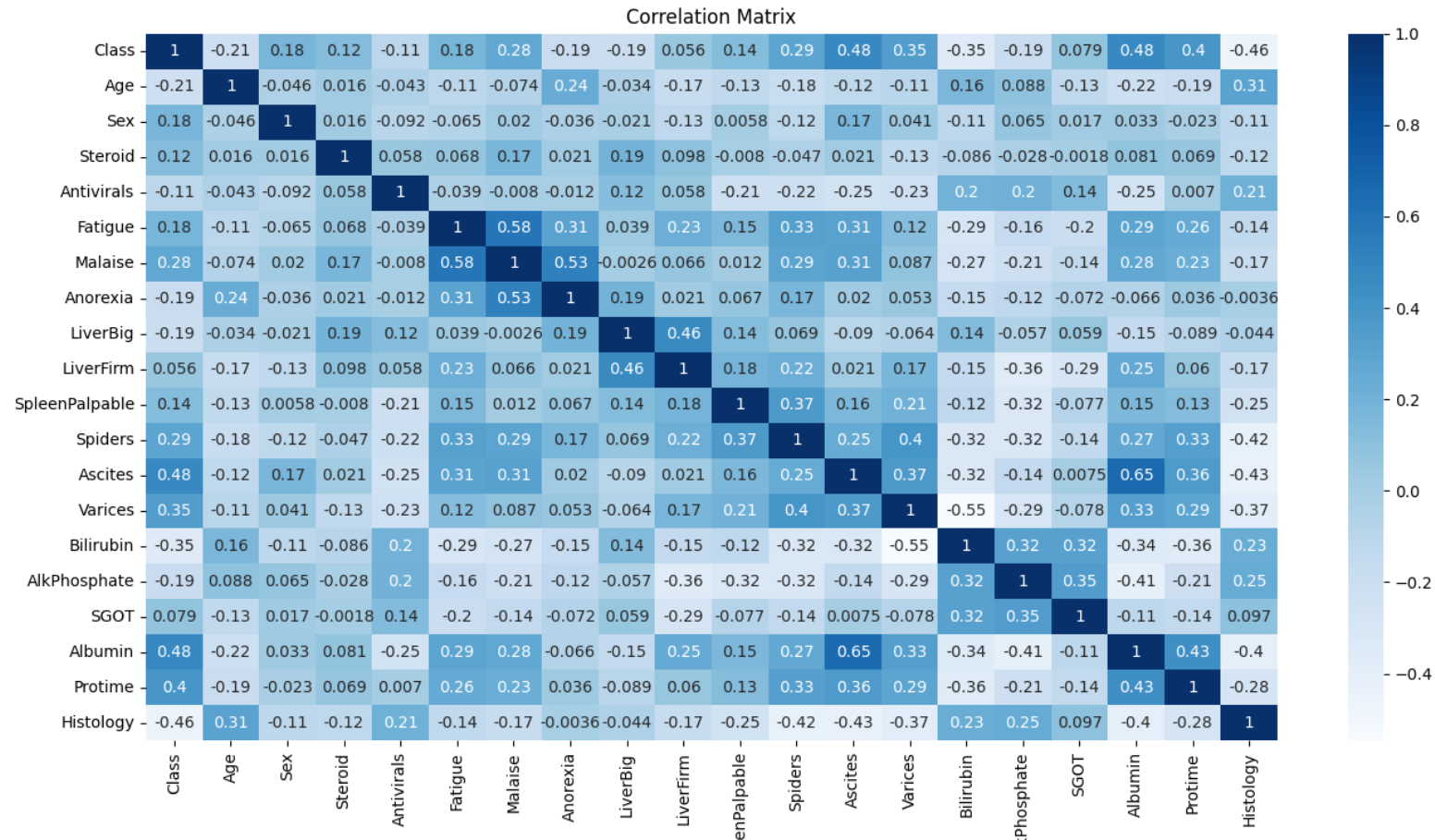
- Box plot, bu özelliklerin dağılımını, merkezi eğilimlerini ve aykırı değerlerini aynı anda gösteren bir grafikdir.
- 'Bilirubin', 'AlkPhosphate' ve 'SGOT' özelliklerinin box plotu
- 'Bilirubin' özelliğinin diğerlerine göre daha yüksek medyan değerine ve daha fazla aykırı değere sahip olduğu görülebilir. 'AlkPhosphate' özelliğinin dağılımının daha simetrik olduğu, 'SGOT' özelliğinin diğerlerine göre daha geniş bir dağılıma sahip

Hepatit Veri Seti Karar Ağacı Modeli

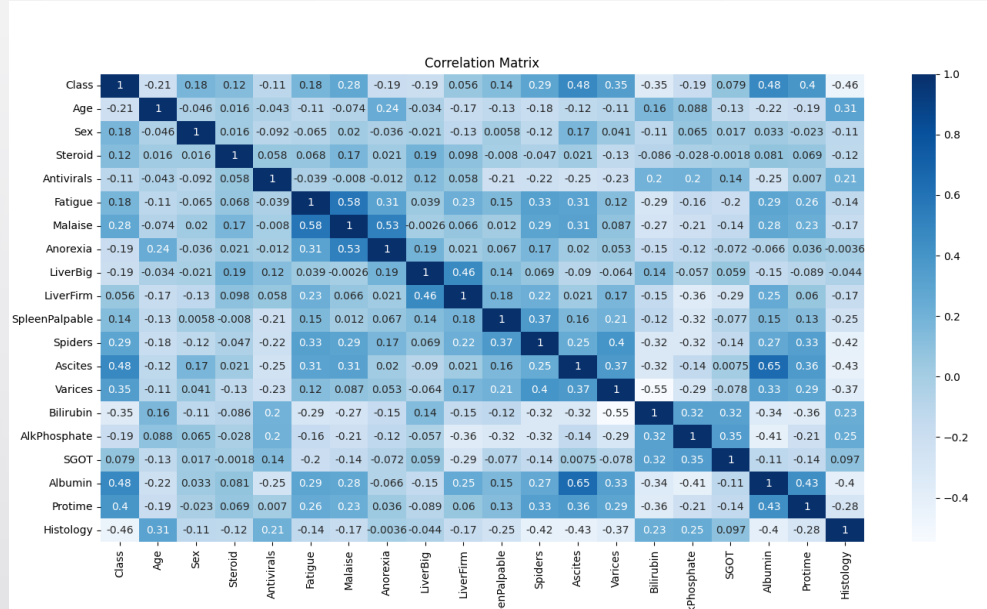


- Bu karar ağacı, karaciğer hastalığı (Hepatit) teşhisi için kullanılan bir makine öğrenimi modelini temsil eder.
- özelliklerin değerlerine dayalı olarak bir kişinin sağlıklı veya hastalıklı olup olmadığını tahmin eder.

Korelasyon Matrisi



Korelasyon Matirisi Analizi



- Korelasyon matrisi, bir veri kümesindeki farklı özellikler (değişkenler) arasındaki ilişkiyi ölçmek için kullanılan bir matrisdir.
- Korelasyon, iki değişken arasındaki ilişki derecesini ölçer ve değeri -1 ile 1 arasında değişir.
- ‘Albumin’ ve ‘Ascites’ arasındaki korelasyon diğer özelliklere göre daha yüksektir.

Diğer Çalışmalar

TABLE VI. *CONFUSION MATRIX FOR DECISION TREES*

	Predicted = YES	Predicted = NO
Actual = YES	6	4
Actual = NO	3	26

TABLE VII. *DECISION TREES*

	Precision	Recall	F1 Score	Support
1.	0.67	0.60	0.63	10
2.	0.87	0.90	0.88	29
Average/Total	0.82	0.82	0.82	39


- Aynı veri seti üzerinde yapılmış başka bir çalışma olan “**Application of Machine Learning Classification Algorithms on Hepatitis Dataset**” isimli çalışma
- %82,05'lik doğruluk oranı ile Karar Ağacı algoritması



TABLE III. ACUURACY MEASURES

Classifier	TPR	FPR	Precision	Recall
Decision Stump	0.838	0.590	0.814	0.838
Hoeffding Tree	0.788	0.661	0.767	0.788
J48	0.863	0.523	0.846	0.863
LMT	0.850	0.401	0.850	0.850
Random Forest	0.875	0.458	0.863	0.875
Random Tree	0.800	0.411	0.825	0.800
REP Tree	0.788	0.723	0.750	0.788

- Hepatit veri seti üzerinde yapılmış olan başka bir çalışma “ **An Empirical Analysis of Decision Tree Algorithms:Modeling Hepatitis Data**” isimli çalışma
- Random Forest Algoritması diğer algoritmalara göre çalışmamdaki sonuçlara daha yakın.



Sonuç

- Modelin doğruluğu (accuracy) %87,5 olarak hesaplanmıştır ve bu sonuç, modelin doğru sınıflandırma oranının yüksek olduğunu göstermektedir.
- Precision, modelin pozitif olarak tahmin ettiği örneklerin gerçekten pozitif olma olasılığını gösterir ve modelin pozitif tahminlerinin %92,86'sının gerçekten pozitif olduğu görülmektedir. Recall, gerçek pozitif örneklerin model tarafından kaçının tespit edildiğini gösterir ve model, gerçek pozitif örneklerin %92,86'sının doğru bir şekilde tespit edildiğini göstermektedir. F1 score'u %92,86'dır,
- Yani model, hem precision hem de recall (hassasiyet) açısından başarılı bir performans sergilemektedir.

Kaynakça

- <https://archive.ics.uci.edu/ml/datasets/hepatitis>
- <file:///C:/Users/saltu/Downloads/live-120-1428-jair.pdf>
- <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=589207&tag=1>
- https://ieeexplore.ieee.org/abstract/document/7275013?casa_token=q-5uAjQvHoMAAAAA:vaUb2qc0RoAokb5srk7wGFalA-BvaYDm20G7iUK_SkLb4T2xPabCcpoKlO7A-GmhBn_bKGZ4hUz0
- https://www.ripublication.com/ijaer18/ijaerv13n16_45.pdf