Veri Madenciliği İle Akciğer Kanser Tespiti

EKİP:

0220020100 6

Samed Sonkaya

0220020101 3

Anıl Berkan Torun

0220020101 1

Habib Şako

Veri Madenciliği İle Akciğer Kanser Tespiti

Veri madenciliği, büyük veri setlerinden anlamlı bilgiler çıkarmak için kullanılan bir analiz yöntemidir.

Veri madenciliği ile akciğer kanseri tespiti; önceden toplanan verilerden hastanın sigara kullanımı, yaşı alkol kullanımı ve yaşam kalitesi parametreleri ile analizler oluşturur. Bu veriler üzerinde yapılan C4.5 algoritması ile potansiyel kanser vakalarını belirlemede kullanılabilir.

Sonuç olarak bu teknoloji, hastalığın erken teşhisini kolaylaştırarak, daha etkili tedavi stratejilerinin belirlenmesine ve hastaların yaşam kalitesinin artırılmasına katkı sağlar.

Kullandığımız Algoritma C4.5

- Karar ağaçları ile sınıflandırma yapmaktadır.
  - ✓ *Ağaçtaki her düğüm bir özellikteki testi gösterir.*
  - ✓ *Düğüm dalları testin sonucunu belirtir.*
  - ✓ *Ağaç yaprakları sınıf etiketlerini içerir.*

- Entropiye dayalı bir sınıflandırma algoritmasıdır.
  - ✓ *Entropi rastgeleliği, belirsizliği ve beklenmeyen durumun ortaya çıkma olasılığını gösterir.*

- ID3 algoritmasından tek farkı nümerik değerlerin kategorik değerler haline dönüştürülmesidir.

Kullanılacak Data Seti Örneği

Name	Surname	Age	Smokes	AreaQ	Alkhol	Result
Yul	Brynner	18	10	6	3	0
Joan	Crawford	25	2	5	1	0
Jane	Wyman	28	20	2	8	1
Anna	Magnani	34	25	4	8	1
Katharine	Hepburn	39	18	8	1	0
Katharine	Hepburn	42	22	3	5	1
Barbra	Streisand	19	12	8	0	0
Maggie	Smith	62	5	4	3	1
Glenda	Jackson	73	10	7	6	1
Jane	Fonda	55	15	1	3	1
Maximilian	Schell	33	8	8	1	0
Gregory	Peck	22	20	6	2	0
Sidney	Poitier	44	5	8	1	0
Rex	Harrison	77	3	2	6	1
Lee	Marvin	21	20	5	3	0
Paul	Scofield	37	15	6	2	0
Rod	Steiger	34	12	8	0	0
John	Wayne	55	20	1	4	1
Gene	Hackman	40	20	2	7	1
Marlon	Brando	36	13	5	2	0

https://www.kaggle.com/datasets/yusufdede/lung-cancer-dataset/data

Verilerin Alınması ve Gruplandırılması (Ön İşlem)

Yapılan işlemler:

- Verilerin .csv dosyasından alınması.
- Veri sütunlarının belirlenen değerler aralığında gruplandırılması. Ör: 0 ile 25 yaş arasındaki değerlere 'Genc' grubuna atanması.
- Oluşturulan yeni değerler ile yeni tablo oluşturulması.

```
data = pd.read_csv('lung_cancer_examples.csv')
print(data)
df = pd.DataFrame(data)
```

```
#Verinin gruplandırılması

df['Age_Class'] = pd.cut(df['Age'], bins=[0, 25, 40, float('inf')], labels=['Genc', 'Orta', 'Yasli'])
df['Alkhol_Class'] = pd.cut(df['Alkhol'], bins=[-1, 3, 6, float('inf')], labels=['0', '1', '2'])
df['Smokes_Class'] = pd.cut(df['Smokes'], bins=[-1, 7, 13, float('inf')], labels=['0', '1', '2'])
df['AreaQ_Class'] = pd.cut(df['AreaQ'], bins=[-1, 3, 7, float('inf')], labels=['0', '1', '2'])

new_data= df[['Age_Class', 'Alkhol_Class','Smokes_Class','AreaQ_Class','Result']]
Entropi_data= df[['Result']]
print(new_data)
```

Örnek

Eşik değerinin belirlenmesi

- Nitelik 2 = {65, 70, 75, **80, 85**, 90, 95, 96} için eşik değer **(80+85)/2 = 83** alınmıştır.

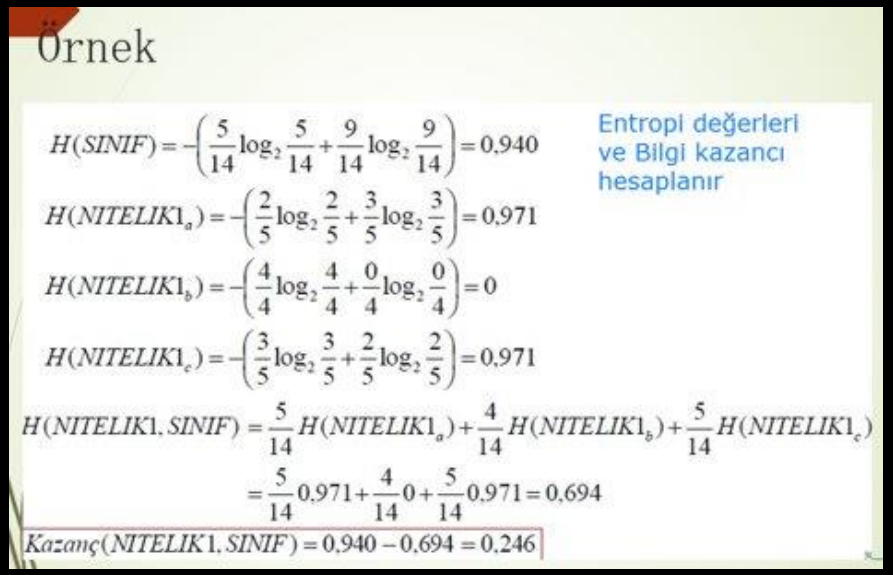
NİTELİK1	NİTELİK2	NİTELİK3	SINIF
a	70	doğru	sınıf1
a	90	doğru	sınıf2
a	85	yanlış	sınıf2
a	95	yanlış	sınıf2
a	70	yanlış	sınıf1
b	90	doğru	sınıf1
b	78	yanlış	sınıf1
b	65	doğru	sınıf1
b	75	yanlış	sınıf1
c	80	doğru	sınıf2
c	70	doğru	sınıf2
c	80	yanlış	sınıf1
c	70	yanlış	sınıf1
c	96	yanlış	sınıf1

NİTELİK2≤ 83
veya
NİTELİK2>83
testi uygulanarak
düzenleme
yapıldığında
yandaki tablo
elde edilir.

Entropilerin Hesaplanması

Yapılanlar İşlemler:

- Gruplama yapılan sınıftaki her bir niteliğe göre ayrı ayrı entropilerin alınması.
- Sınıfın entropisinin bulunması.
- Entropi_result değerinden sınıf entropi değerinin çıkarılarak kazancın bulunması.




```
################### Age Kazanç Hesaplanması#############################

Age0 = new_data[new_data['Age_Class'] == 'Genc']
Age1 = new_data[new_data['Age_Class'] == 'Orta']
Age2 = new_data[new_data['Age_Class'] == 'Yasli']

Age0_result= Age0[['Result']]
Age0_entropy=calculate_entropy(Age0_result)

Age1_result= Age1[['Result']]
Age1_entropy=calculate_entropy(Age1_result)

Age2_result= Age2[['Result']]
Age2_entropy=calculate_entropy(Age2_result)

kazanc_Age= Entropi_result-((len(Age0)*Age0_entropy
                            +len(Age1)*Age1_entropy
                            +len(Age2)*Age2_entropy)/(len(Age0)+len(Age1)+len(Age2)))
print("Kazanc_Age = ",kazanc_Age)
```

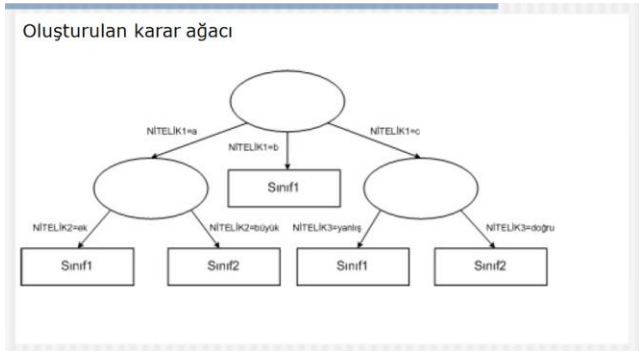
```
58      Yasli              0              2
Result_Entropisi =  0.9981341775041116
Kazanc_Age =  0.31959267094489074
Kazanc_Smoke =  0.18539078304824685
Kazanc_Alkhol =  0.6012021187969507
Kazanc_AreaQ =  0.5305682824370093
```

Bulunan Entropilere Göre Karar Ağacının Oluşturulması ve Test Edilmesi

Yapılan İşlemler:

- Bulunan kazanç değerlerine göre en büyük olan değerden en küçük olan değere doğru sıralama yapılır.
- Yapılan sıralamaya göre en büyük değerden başlanarak karar ağacı oluşturulmaya başlanır.
- Dallanmada bulunan değerlerin hepsi aynı sınıfa aitse düğüm yaprak olarak sonlanıyor ve sınıf etiketini alıyor. (0,1)
- Ağaç oluşturulduktan sonra test için örnek veri gönderiliyor ve tahmin değeri alınıyor.

```
Kazanc_AreaQ =  0.5305682824370093
|--- Alkhol_Class <= 0.50
|   |--- AreaQ_Class <= 0.50
|   |   |--- class: 1
|   |--- AreaQ_Class >  0.50
|   |   |--- Age_Class <= 1.50
|   |   |   |--- class: 0
|   |   |--- Age_Class >  1.50
|   |   |   |--- AreaQ_Class <= 1.50
|   |   |   |   |--- Smokes_Class <= 1.50
|   |   |   |   |   |--- Smokes_Class <= 0.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- Smokes_Class >  0.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- Smokes_Class >  1.50
|   |   |   |   |   |--- class: 1
|   |   |   |--- AreaQ_Class >  1.50
|   |   |   |   |--- class: 0
|--- Alkhol_Class >  0.50
|   |--- AreaQ_Class <= 1.50
```




```
class_mapping = {'Genc': 0, 'Orta': 1, 'Yasli': 2}
new_data['Age_Class'] = df['Age_Class'].map(class_mapping)

X = new_data[['Alkhol_Class', 'Smokes_Class', 'AreaQ_Class','Age_Class']]
y = new_data['Result']

# Eğitim ve test veri setlerine ayırma
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Karar ağacı modeli oluşturma
clf = DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)
tree_rules = export_text(clf, feature_names=['Alkhol_Class', 'Smokes_Class', 'AreaQ_Class', 'Age_Class'])
print(tree_rules)
```

```
# Karar ağacından test etme
new_data_test = pd.DataFrame({'Alkhol_Class': [0], 'Smokes_Class': [2], 'AreaQ_Class': [0], 'Age_Class':[2]})
prediction = clf.predict(new_data_test)
```

```
|   |--- AreaQ_Class >  1.50
|   |   |--- class: 0

Tahmin: 1
```

Teşekkürler