Bölüm 11: Algoritmik Adalet ve Etik

Algoritmik Oyun Kuramı

Karışıklık matrisi | Demografik eşitlik | Eşitlenmiş olasılıklar | Tahmin eşitliği | Kalibrasyon | Uyumsuzluk teoremleri | Kes-ve-seç | Orantılılık ve kıskançsızlık | Etik ilkeler ve denetim

Sercan KÜLCÜ · Giresun Üniversitesi · Bilgisayar Mühendisliği

© Sercan Külcü. Tüm hakları saklıdır. · Sunumun etkileşimli HTML sürümü: https://sercankulcu.github.io/files/agt/slides/Bolum_11_Algoritmik_Adalet_ve_Etik.html

GİRİŞ

Oyunla Başlangıç: Adil Bölüşüm Oyunu

Oyunun kuralı

İki oyuncu bölünemez malları sırayla birer birer seçer; her malın kartında iki oyuncunun o mala verdiği değer yazar. Oyunun sabit örnek turunda (Example) sağdaki dört mal kullanılır; diğer seviyelerde değerler her turda rastgele çekilir.

Merak sorusu

- Rakip hep kendi en sevdiği malı alıyorsa, siz de hep en değerlisini mi almalısınız?
- Herkesin kendi payını ötekininkinden değerli bulduğu bir paylaşım her zaman var mı?

Pasta	Kitap	Kulaklık	Top
O1: 5	O1: 4	O1: 3	O1: 1
O2: 1	O2: 9	O2: 8	O2: 1

Sıra: O1, O2, O1, O2

O2 (bilgisayar) her seferinde kendi gözünde en değerli kalan malı alır.

Adil Bölüşüm Oyunu →

GİRİŞ

Öğrenme Çıktıları

Ölçütleri hesaplamak

İki grubun karışıklık matrisinden demografik eşitlik, eşitlenmiş olasılıklar, tahmin eşitliği ve kalibrasyonu hesaplamak.

Uyumsuzluğu göstermek

Taban oranları farklıyken ölçütlerin neden birlikte sağlanamadığını sayısal olarak göstermek.

Adil bölüşüm

Kes-ve-seç yöntemini uygulamak; orantılılık ve kıskançsızlığı ayırt etmek.

Etik ve teşvikler

Şeffaflık, hesap verebilirlik, denetim ve stratejik davranışın tasarıma etkisini tartışmak.

GİRİŞ

Algoritmik Kararlar Neden Önemli?

- Algoritmalar artık yüksek etkili kararlar veriyor ya da önerisiyle yönlendiriyor: **kredi onayı, işe alım, sağlık teşhisi, sosyal medya önerileri, ceza adaleti**.
- Adalet ve etik ihlalleri bireylere zarar verir ve kurumlara, teknolojiye duyulan **toplumsal güveni** aşındırır.
- Oyun kuramı bağlantısı: karar kuralı bir **mekanizmadır**; farklı paydaşların (başvuran, kurum, toplum) çıkarları çatışır.

Amazon işe alım (2018)

Geçmiş özgeçmişlerle eğitilen model "kadın" kelimesi geçen başvuruları cezalandırdığı için kullanımdan kaldırıldı.

COMPAS (2016)

Yeniden suç işleme risk puanı: gruplar arası hata oranı farkı tartışması: taban oranları farklıyken adalet ölçütleri çatışır.

Kredi skorlama

Aynı puan herkes için aynı riski mi gösteriyor? Kimler hiç kredi alamıyor?

GİRİŞ

Önyargı ve Ayrımcılık Nereden Gelir?

Tarihsel veri

Geçmişteki eşitsizlikler etiketlere yansır; model onları "doğru" diye öğrenir (işe alımda cinsiyet önyargısı).

Temsil eksikliği

Bir grup veride az ise model o grupta daha çok hata yapar (yüz tanımada koyu tenli kadınlarda en yüksek hata: Gender Shades, 2018).

Ölçüm ve vekil değişken

Hedef ("sağlık ihtiyacı") yerine vekil ("sağlık harcaması") ölçülürse erişimi az olan grup düşük görünür.

Tasarım ve geri besleme

Yanlış optimizasyon hedefi, korunan özelliğin vekilleri (posta kodu) ve kararların gelecekteki veriyi değiştirmesi.

Korunan özelliği (cinsiyet, etnik köken) modelden çıkarmak yetmez: diğer değişkenler onu dolaylı olarak taşır. Bu yüzden sonuçları **grup bazında ölçmek** gerekir.

Karışıklık Matrisi (Confusion Matrix)

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP **40**	FN **10**
ödemez ($Y=0$)	FP **10**	TN **40**

Kredi örneği: Y = gerçekte geri öder mi, $\hat{Y}$ = model onaylar mı; $n = TP + FP + FN + TN$

Oran	Tanım	Soru
Seçim oranı	(TP + FP) / n	Kaçta kaçı onay alıyor?
TPR (doğru pozitif oranı)	TP / (TP + FN)	Ödeyecek olanların kaçı onaylandı?
FPR (yanlış pozitif oranı)	FP / (FP + TN)	Ödemeyecek olanların kaçı onaylandı?
PPV (kesinlik)	TP / (TP + FP)	Onaylananların kaçı ödüyor?
Taban oranı p	(TP + FN) / n	Grupta ödeyenlerin payı

Her adalet ölçütü bu oranlardan birinin **gruplar arasında eşit** olmasını ister. Hangisinin eşitleneceği bir değer seçimidir.

Örnek: İki Grup, Tek Model

grup	puan	kişi	öder	ödemez	karar (eşik 0,5)
A	0,80	50	40	10	onay
A	0,20	50	10	40	ret
B	0,80	20	16	4	onay
B	0,20	80	16	64	ret

Her grupta 100 başvuran; model iki puan veriyor: 0,8 ("düşük risk") ve 0,2 ("yüksek risk")

A grubunda ödeyenlerin payı (taban oranı) %50, B grubunda %32. Model 0,5 eşiğiyle karar veriyor.

Örnek: İki Grup, Tek Model

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 40	FN 10
ödemez ($Y=0$)	FP 10	TN 40

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 16	FN 16
ödemez ($Y=0$)	FP 4	TN 64

Puan 0,8 olanlar onaylanır: A'da 40 öder (TP) + 10 ödemez (FP); B'de 16 TP + 4 FP.

Örnek: İki Grup, Tek Model

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 40	FN 10
ödemez ($Y=0$)	FP 10	TN 40

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 16	FN 16
ödemez ($Y=0$)	FP 4	TN 64

Puan 0,2 olanlar reddedilir: A'da 10 ödeyecek kişi reddedilir (FN), 40 doğru ret (TN); B'de 16 FN + 64 TN.

Örnek: İki Grup, Tek Model

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP **40**	FN **10**
ödemez ($Y=0$)	FP **10**	TN **40**

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP **16**	FN **16**
ödemez ($Y=0$)	FP **4**	TN **64**

İki karışıklık matrisi hazır. Doğruluk (accuracy) iki grupta da %80 ((40+40)/100 ve (16+64)/100). Model "adil" mi? Ölçütlere göre tek tek bakalım.

Demografik Eşitlik Adım Adım

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 40	FN 10
ödemez ($Y=0$)	FP 10	TN 40

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 16	FN 16
ödemez ($Y=0$)	FP 4	TN 64

Demografik eşitlik (demographic parity): $P(\hat{Y}=1 \mid A) = P(\hat{Y}=1 \mid B)$ olmalı. Onaylananlar: A'da 40 + 10 = 50, B'de 16 + 4 = 20.

Demografik Eşitlik Adım Adım

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 40	FN 10
ödemez ($Y=0$)	FP 10	TN 40

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 16	FN 16
ödemez ($Y=0$)	FP 4	TN 64

Seçim oranları: A = 50/100 = 0,50, B = 20/100 = 0,20. Fark 0,30 ⇒ demografik eşitlik **sağlanmıyor**.

Demografik Eşitlik Adım Adım

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 40	FN 10
ödemez ($Y=0$)	FP 10	TN 40

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 16	FN 16
ödemez ($Y=0$)	FP 4	TN 64

Oran: 0,20 / 0,50 = 0,40 < 0,8. ABD istihdam hukukundaki "beşte dört kuralı" (disparate impact) da ihlal edilir.

Demografik Eşitlik Adım Adım

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 40	FN 10
ödemez ($Y=0$)	FP 10	TN 40

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 16	FN 16
ödemez ($Y=0$)	FP 4	TN 64

Ama gruplar farklı: gerçekten ödeyenler A'da 50, B'de 32. Demografik eşitliği zorlamak ya B'de ödemeyecek kişileri onaylamak ya da A'da ödeyecekleri reddetmek demektir.

Eşitlenmiş Olasılıklar Adım Adım

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP **40**	FN **10**
ödemez ($Y=0$)	FP **10**	TN **40**

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP **16**	FN **16**
ödemez ($Y=0$)	FP **4**	TN **64**

Eşitlenmiş olasılıklar (equalized odds; Hardt vd., 2016): gerçek sonuç koşulu altında hata oranları eşit olmalı: $TPR_A = TPR_B$ ve $FPR_A = FPR_B$.

Eşitlenmiş Olasılıklar Adım Adım

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 40	FN 10
ödemez ($Y=0$)	FP 10	TN 40

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 16	FN 16
ödemez ($Y=0$)	FP 4	TN 64

TPR (ödeyecek olanlar arasında onay): A = 40/(40+10) = 0,80, B = 16/(16+16) = 0,50. B'nin kredi değerli başvuranlarının yarısı reddediliyor!

Eşitlenmiş Olasılıklar Adım Adım

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 40	FN 10
ödemez ($Y=0$)	FP 10	TN 40

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 16	FN 16
ödemez ($Y=0$)	FP 4	TN 64

FPR (ödemeyecekler arasında onay): A = 10/(10+40) = 0,20, B = 4/(4+64) = 0,059.

Eşitlenmiş Olasılıklar Adım Adım

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 40	FN 10
ödemez ($Y=0$)	FP 10	TN 40

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 16	FN 16
ödemez ($Y=0$)	FP 4	TN 64

TPR farkı 0,30, FPR farkı 0,141 ⇒ eşitlenmiş olasılıklar **sağlanmıyor**. Yalnız TPR eşitliği istenirse adı **fırsat eşitliği**dir (equal opportunity).

Tahmin Eşitliği Adım Adım

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 40	FN 10
ödemez ($Y=0$)	FP 10	TN 40

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 16	FN 16
ödemez ($Y=0$)	FP 4	TN 64

Tahmin eşitliği (predictive parity): onay alanların gerçekten ödeme oranı (PPV) gruplar arasında eşit olmalı. Bakış açısı: kararı kullanan kurum.

Tahmin Eşitliği Adım Adım

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP **40**	FN **10**
ödemez ($Y=0$)	FP **10**	TN **40**

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP **16**	FN **16**
ödemez ($Y=0$)	FP **4**	TN **64**

PPV: A = 40/(40+10) = 0,80, B = 16/(16+4) = 0,80 ⇒ tahmin eşitliği **sağlanıyor**.

Tahmin Eşitliği Adım Adım

Grup A

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 40	FN 10
ödemez ($Y=0$)	FP 10	TN 40

Grup B

	onay ($\hat{Y}=1$)	ret ($\hat{Y}=0$)
öder ($Y=1$)	TP 16	FN 16
ödemez ($Y=0$)	FP 4	TN 64

Ret tarafında da (NPV: reddedilenlerin gerçekten ödemeyen payı = TN/(TN + FN)): A = 40/50 = 0,80, B = 64/80 = 0,80. "Onay" ve "ret" etiketleri iki grupta aynı anlamı taşıyor.

Kalibrasyon Adım Adım

grup	puan s	kişi	öder	gerçekleşme oranı	s ile aynı mı?
A	0,80	50	40	?	
A	0,20	50	10	?	
B	0,80	20	16	?	
B	0,20	80	16	?	

Kalibrasyon (calibration within groups): puanı s olan kişilerin gerçekten ödeme oranı, **her grupta** s olmalı. Puan, grup ne olursa olsun aynı riski anlatmalı.

Kalibrasyon Adım Adım

grup	puan s	kişi	öder	gerçekleşme oranı	s ile aynı mı?
A	**0,80**	**50**	**40**	**40/50 = 0,80**	**evet**
A	0,20	50	10	?	
B	0,80	20	16	?	
B	0,20	80	16	?	

A grubu, puan 0,8: 50 kişiden 40'ı öder ⇒ 0,80.

Kalibrasyon Adım Adım

grup	puan s	kişi	öder	gerçekleşme oranı	s ile aynı mı?
A	0,80	50	40	40/50 = 0,80	**evet**
A	**0,20**	**50**	**10**	**10/50 = 0,20**	**evet**
B	0,80	20	16	?	
B	0,20	80	16	?	

A grubu, puan 0,2: 50 kişiden 10'u öder ⇒ 0,20.

Kalibrasyon Adım Adım

grup	puan s	kişi	öder	gerçekleşme oranı	s ile aynı mı?
A	0,80	50	40	40/50 = 0,80	**evet**
A	0,20	50	10	10/50 = 0,20	**evet**
B	**0,80**	**20**	**16**	**16/20 = 0,80**	**evet**
B	0,20	80	16	?	

B grubu, puan 0,8: 20 kişiden 16'sı öder ⇒ 0,80.

Kalibrasyon Adım Adım

grup	puan s	kişi	öder	gerçekleşme oranı	s ile aynı mı?
A	0,80	50	40	40/50 = 0,80	**evet**
A	0,20	50	10	10/50 = 0,20	**evet**
B	0,80	20	16	16/20 = 0,80	**evet**
B	**0,20**	**80**	**16**	**16/80 = 0,20**	**evet**

B grubu, puan 0,2: 80 kişiden 16'sı öder ⇒ 0,20. Model iki grupta da **kalibre**: puan dürüst bir olasılıktır.

Kod İzleme: `olcutler(TP, FP, FN, TN)`

Grup A

	onay (Ŷ=1)	ret (Ŷ=0)
öder (Y=1)	TP 40	FN 10
ödemez (Y=0)	FP 10	TN 40

A grubunun matrisiyle fonksiyon çağrılır.

Adım 1 / 11

```
1  def olcutler(TP, FP, FN, TN):
2    n = TP + FP + FN + TN
3    secim = (TP + FP) / n    # demografik eşitlik
4    tpr = TP / (TP + FN)     # eşitlenmiş olasılıklar
5    fpr = FP / (FP + TN)
6    ppv = TP / (TP + FP)     # tahmin eşitliği
7    return secim, tpr, fpr, ppv
8
▶ A = olcutler(40, 10, 10, 40)
10 B = olcutler(16, 4, 16, 64)
11 ADLAR = ["seçim", "TPR", "FPR", "PPV"]
12 for i, ad in enumerate(ADLAR):
13   print(ad, round(A[i], 3), round(B[i], 3))
```

DEĞİŞKENLER

—

ÇIKTI

Kod İzleme: `olcutler(TP, FP, FN, TN)`

Grup A

	onay (Ŷ=1)	ret (Ŷ=0)
öder (Y=1)	TP 40	FN 10
ödemez (Y=0)	FP 10	TN 40

n = 40 + 10 + 10 + 40 = 100 başvuran.

Adım 2 / 11

```
▶ def olcutler(TP, FP, FN, TN):
▶     n = TP + FP + FN + TN
3     secim = (TP + FP) / n    # demografik eşitlik
4     tpr = TP / (TP + FN)     # eşitlenmiş olasılıklar
5     fpr = FP / (FP + TN)
6     ppv = TP / (TP + FP)     # tahmin eşitliği
7     return secim, tpr, fpr, ppv
8
9   A = olcutler(40, 10, 10, 40)
10  B = olcutler(16, 4, 16, 64)
11  ADLAR = ["seçim", "TPR", "FPR", "PPV"]
12  for i, ad in enumerate(ADLAR):
13      print(ad, round(A[i], 3), round(B[i], 3))
```

DEĞİŞKENLER

`TP = 40` `FP = 10` `FN = 10` `TN = 40` `n = 100`

ÇIKTI

Kod İzleme: `olcutler(TP, FP, FN, TN)`

Grup A

	onay (Ŷ=1)	ret (Ŷ=0)
öder (Y=1)	TP 40	FN 10
ödemez (Y=0)	FP 10	TN 40

Seçim oranı = (40 + 10) / 100 = 0,5 (demografik eşitlikte karşılaştırılır).

Adım 3 / 11

```
1  def olcutler(TP, FP, FN, TN):
2    n = TP + FP + FN + TN
▶    secim = (TP + FP) / n    # demografik eşitlik
4    tpr = TP / (TP + FN)     # eşitlenmiş olasılıklar
5    fpr = FP / (FP + TN)
6    ppv = TP / (TP + FP)     # tahmin eşitliği
7    return secim, tpr, fpr, ppv
8
9  A = olcutler(40, 10, 10, 40)
10 B = olcutler(16, 4, 16, 64)
11 ADLAR = ["seçim", "TPR", "FPR", "PPV"]
12 for i, ad in enumerate(ADLAR):
13   print(ad, round(A[i], 3), round(B[i], 3))
```

DEĞİŞKENLER

`TP = 40` `FP = 10` `FN = 10` `TN = 40` `n = 100` `secim = 0.5`

ÇIKTI

Kod İzleme: `olcutler(TP, FP, FN, TN)`

Grup A

	onay (Ŷ=1)	ret (Ŷ=0)
öder (Y=1)	TP 40	FN 10
ödemez (Y=0)	FP 10	TN 40

TPR = 40 / (40 + 10) = 0,8.

Adım 4 / 11

```
1  def olcutler(TP, FP, FN, TN):
2    n = TP + FP + FN + TN
3    secim = (TP + FP) / n    # demografik eşitlik
▶    tpr = TP / (TP + FN)     # eşitlenmiş olasılıklar
5    fpr = FP / (FP + TN)
6    ppv = TP / (TP + FP)     # tahmin eşitliği
7    return secim, tpr, fpr, ppv
8
9  A = olcutler(40, 10, 10, 40)
10 B = olcutler(16, 4, 16, 64)
11 ADLAR = ["seçim", "TPR", "FPR", "PPV"]
12 for i, ad in enumerate(ADLAR):
13   print(ad, round(A[i], 3), round(B[i], 3))
```

DEĞİŞKENLER

`TP = 40` `FP = 10` `FN = 10` `TN = 40` `n = 100` `secim = 0.5` `tpr = 0.8`

ÇIKTI

Kod İzleme: `olcutler(TP, FP, FN, TN)`

Grup A

	onay (Ŷ=1)	ret (Ŷ=0)
öder (Y=1)	TP 40	FN 10
ödemez (Y=0)	FP 10	TN 40

FPR = 10 / (10 + 40) = 0,2.

Adım 5 / 11

```
1  def olcutler(TP, FP, FN, TN):
2    n = TP + FP + FN + TN
3    secim = (TP + FP) / n    # demografik eşitlik
4    tpr = TP / (TP + FN)     # eşitlenmiş olasılıklar
▶    fpr = FP / (FP + TN)
6    ppv = TP / (TP + FP)     # tahmin eşitliği
7    return secim, tpr, fpr, ppv
8
9  A = olcutler(40, 10, 10, 40)
10 B = olcutler(16, 4, 16, 64)
11 ADLAR = ["seçim", "TPR", "FPR", "PPV"]
12 for i, ad in enumerate(ADLAR):
13   print(ad, round(A[i], 3), round(B[i], 3))
```

DEĞİŞKENLER

`TP = 40` `FP = 10` `FN = 10`
`TN = 40` `n = 100` `secim = 0.5`
`tpr = 0.8` `fpr = 0.2`

ÇIKTI

Kod İzleme: `olcutler(TP, FP, FN, TN)`

Grup A

	onay (Ŷ=1)	ret (Ŷ=0)
öder (Y=1)	TP 40	FN 10
ödemez (Y=0)	FP 10	TN 40

PPV = 40 / (40 + 10) = 0,8.

Adım 6 / 11

```
1  def olcutler(TP, FP, FN, TN):
2    n = TP + FP + FN + TN
3    secim = (TP + FP) / n    # demografik eşitlik
4    tpr = TP / (TP + FN)     # eşitlenmiş olasılıklar
5    fpr = FP / (FP + TN)
▶    ppv = TP / (TP + FP)     # tahmin eşitliği
7    return secim, tpr, fpr, ppv
8
9  A = olcutler(40, 10, 10, 40)
10 B = olcutler(16, 4, 16, 64)
11 ADLAR = ["seçim", "TPR", "FPR", "PPV"]
12 for i, ad in enumerate(ADLAR):
13   print(ad, round(A[i], 3), round(B[i], 3))
```

DEĞİŞKENLER

`TP = 40` `FP = 10` `FN = 10`
`TN = 40` `n = 100` `secim = 0.5`
`tpr = 0.8` `fpr = 0.2` `ppv = 0.8`

ÇIKTI

Kod İzleme: `olcutler(TP, FP, FN, TN)`

Grup A

	onay (Ŷ=1)	ret (Ŷ=0)
öder (Y=1)	TP 40	FN 10
ödemez (Y=0)	FP 10	TN 40

Dört oran bir Python demeti (tuple) olarak döner ve A'ya atanır.

Adım 7 / 11

```
1  def olcutler(TP, FP, FN, TN):
2    n = TP + FP + FN + TN
3    secim = (TP + FP) / n    # demografik eşitlik
4    tpr = TP / (TP + FN)     # eşitlenmiş olasılıklar
5    fpr = FP / (FP + TN)
6    ppv = TP / (TP + FP)     # tahmin eşitliği
▶    return secim, tpr, fpr, ppv
8
▶  A = olcutler(40, 10, 10, 40)
10 B = olcutler(16, 4, 16, 64)
11 ADLAR = ["seçim", "TPR", "FPR", "PPV"]
12 for i, ad in enumerate(ADLAR):
13   print(ad, round(A[i], 3), round(B[i], 3))
```

DEĞİŞKENLER

`A = (0.5, 0.8, 0.2, 0.8)`

ÇIKTI

Kod İzleme: `olcutler(TP, FP, FN, TN)`

Grup B

	onay (Ŷ=1)	ret (Ŷ=0)
öder (Y=1)	TP 16	FN 16
ödemez (Y=0)	FP 4	TN 64

B için aynı hesap: seçim 20/100, TPR 16/32, FPR 4/68 ≈ 0,059, PPV 16/20.

Adım 8 / 11

```
1  def olcutler(TP, FP, FN, TN):
▶    n = TP + FP + FN + TN
▶    secim = (TP + FP) / n    # demografik eşitlik
▶    tpr = TP / (TP + FN)     # eşitlenmiş olasılıklar
▶    fpr = FP / (FP + TN)
▶    ppv = TP / (TP + FP)     # tahmin eşitliği
7    return secim, tpr, fpr, ppv
8
9  A = olcutler(40, 10, 10, 40)
10 B = olcutler(16, 4, 16, 64)
11 ADLAR = ["seçim", "TPR", "FPR", "PPV"]
12 for i, ad in enumerate(ADLAR):
13   print(ad, round(A[i], 3), round(B[i], 3))
```

DEĞİŞKENLER

`TP = 16` `FP = 4` `FN = 16` `TN = 64` `n = 100`

ÇIKTI

Kod İzleme: `olcutler(TP, FP, FN, TN)`

Grup B

	onay (Ŷ=1)	ret (Ŷ=0)
öder (Y=1)	TP 16	FN 16
ödemez (Y=0)	FP 4	TN 64

B = (0.2, 0.5, 0.0588..., 0.8).

Adım 9 / 11

```
1  def olcutler(TP, FP, FN, TN):
2    n = TP + FP + FN + TN
3    secim = (TP + FP) / n    # demografik eşitlik
4    tpr = TP / (TP + FN)     # eşitlenmiş olasılıklar
5    fpr = FP / (FP + TN)
6    ppv = TP / (TP + FP)     # tahmin eşitliği
7    return secim, tpr, fpr, ppv
8
9  A = olcutler(40, 10, 10, 40)
▶ B = olcutler(16, 4, 16, 64)
11 ADLAR = ["seçim", "TPR", "FPR", "PPV"]
12 for i, ad in enumerate(ADLAR):
13   print(ad, round(A[i], 3), round(B[i], 3))
```

DEĞİŞKENLER

`A = (0.5, 0.8, 0.2, 0.8)`

`B = (0.2, 0.5, 0.059, 0.8)`

ÇIKTI

Kod İzleme: `olcutler(TP, FP, FN, TN)`

Grup A

	onay (Ŷ=1)	ret (Ŷ=0)
öder (Y=1)	TP 40	FN 10
ödemez (Y=0)	FP 10	TN 40

Demografik eşitlik yok: 0,5 ≠ 0,2.

Adım 10 / 11

```
1  def olcutler(TP, FP, FN, TN):
2    n = TP + FP + FN + TN
3    secim = (TP + FP) / n    # demografik eşitlik
4    tpr = TP / (TP + FN)     # eşitlenmiş olasılıklar
5    fpr = FP / (FP + TN)
6    ppv = TP / (TP + FP)     # tahmin eşitliği
7    return secim, tpr, fpr, ppv
8
9  A = olcutler(40, 10, 10, 40)
10 B = olcutler(16, 4, 16, 64)
▶  ADLAR = ["seçim", "TPR", "FPR", "PPV"]
▶  for i, ad in enumerate(ADLAR):
▶    print(ad, round(A[i], 3), round(B[i], 3))
```

DEĞİŞKENLER

`i = 0` `ad = seçim`

ÇIKTI

```
seçim 0.5 0.2
```

Kod İzleme: `olcutler(TP, FP, FN, TN)`

Grup B

	onay (Ŷ=1)	ret (Ŷ=0)
öder (Y=1)	TP 16	FN 16
ödemez (Y=0)	FP 4	TN 64

i = 1, 2, 3: TPR farklı ⇒ fırsat eşitliği yok; FPR de farklı ⇒ eşitlenmiş olasılıklar yok; PPV eşit ⇒ tahmin eşitliği var.

Adım 11 / 11

```
1  def olcutler(TP, FP, FN, TN):
2    n = TP + FP + FN + TN
3    secim = (TP + FP) / n    # demografik eşitlik
4    tpr = TP / (TP + FN)     # eşitlenmiş olasılıklar
5    fpr = FP / (FP + TN)
6    ppv = TP / (TP + FP)     # tahmin eşitliği
7    return secim, tpr, fpr, ppv
8
9  A = olcutler(40, 10, 10, 40)
10 B = olcutler(16, 4, 16, 64)
11 ADLAR = ["seçim", "TPR", "FPR", "PPV"]
▶  for i, ad in enumerate(ADLAR):
▶    print(ad, round(A[i], 3), round(B[i], 3))
```

DEĞİŞKENLER

`i = 3` `ad = PPV`

ÇIKTI

```
seçim 0.5 0.2
TPR 0.8 0.5
FPR 0.2 0.059
PPV 0.8 0.8
```

ADALET ÖLÇÜTLERİ

Beş Ölçüt, Tek Model: Sonuç Tablosu

Ölçüt	Eşit olması gereken	A	B	Sağlanıyor mu?
Demografik eşitlik	$P(\hat{Y}=1)$	0,50	0,20	**hayır**
Fırsat eşitliği	$TPR = P(\hat{Y}=1 \mid Y=1)$	0,80	0,50	**hayır**
Eşitlenmiş olasılıklar	TPR ve $FPR = P(\hat{Y}=1 \mid Y=0)$	0,80; 0,20	0,50; 0,059	**hayır**
Tahmin eşitliği	$PPV = P(Y=1 \mid \hat{Y}=1)$	0,80	0,80	**evet**
Kalibrasyon	$P(Y=1 \mid \text{puan} = s) = s$	0,8→0,8; 0,2→0,2	0,8→0,8; 0,2→0,2	**evet**

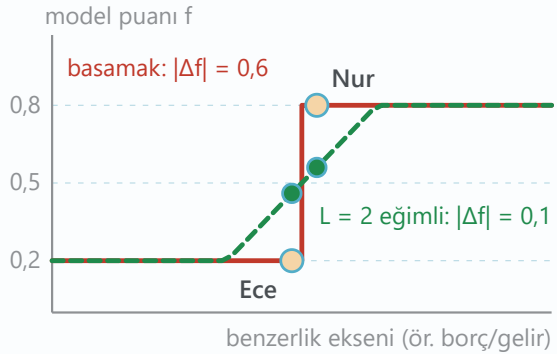
Başvuranın bakışı

"Ödeyebilecek biri olarak onay alma şansım grubuma bağlı mı?" ⇒ TPR / FPR.

Kurumun bakışı

"Onay verdiğim kişinin ödeme olasılığı grubuna bağlı mı?" ⇒ PPV, kalibrasyon.

Grup Adaletinin Ötesi: Bireysel Adalet



Ece ile Nur arasındaki uzaklık d = 0,05; L·d = 0,1

- **Dwork vd. (2012)**: "benzer kişilere benzer davranılmalı": $|f(x) - f(y)| \leq L \cdot d(x, y)$ (Lipschitz koşulu). f(x): x kişisinin model puanı; d(x, y): iki kişinin birbirinden ne kadar farklı olduğu; L: izin verilen en büyük eğim. Yani iki kişinin puan farkı, aralarındaki farkın L katını aşmamalı.
- Kırmızı basamak model eşiğin iki yanındaki Ece ve Nur'a 0,2 ve 0,8 verir: $|\Delta f| = 0,6 > 0,1$ ⇒ **ihlal**. Yeşil model koşulu sağlar.
- Grup ölçütleri **ortalamaları** eşitler: B'den rastgele kişileri onaylamak demografik eşitliği sağlar ama bireysel olarak keyfîdir.

Zorluk: uzaklık ölçüsü d'yi kim belirler? "Kimler benzer?" sorusu da bir değer yargısıdır; bireysel adalet sorunu ortadan kaldırmaz, onu d'nin seçimine taşır.

Ölçütler Neden Birlikte Sağlanamaz?

p = taban oranı, n kişi ⇒ TP = p·n·TPR, FP = (1 − p)·n·FPR

Chouldechova (2017): ölçütler aynı dört sayıdan türetildiği için birbirine bağlıdır. Taban oranı p ile başlayalım.

Ölçütler Neden Birlikte Sağlanamaz?

p = taban oranı, n kişi ⇒ $TP = p \cdot n \cdot TPR$, $FP = (1 - p) \cdot n \cdot FPR$

$PPV = TP / (TP + FP) = p \cdot TPR / (p \cdot TPR + (1 - p) \cdot FPR)$

Tahmin eşitliğinin konusu PPV'yi TPR, FPR ve p cinsinden yazalım.

Ölçütler Neden Birlikte Sağlanamaz?

p = taban oranı, n kişi ⇒ TP = p·n·TPR, FP = (1 − p)·n·FPR

PPV = TP / (TP + FP) = p·TPR / (p·TPR + (1 − p)·FPR)

Paydayla çarp: PPV·[p·TPR + (1 − p)·FPR] = p·TPR ⇒ (1 − p)·FPR·PPV = p·TPR·(1 − PPV)

Düzenle: **FPR = p / (1 − p) · (1 − PPV) / PPV · TPR**

FPR'yi yalnız bırakırsak üç oranı (FPR, TPR, PPV) ve taban oranını bağlayan **özdeşlik** çıkar.

Ölçütler Neden Birlikte Sağlanamaz?

p = taban oranı, n kişi ⇒ TP = p·n·TPR, FP = (1 − p)·n·FPR

PPV = TP / (TP + FP) = p·TPR / (p·TPR + (1 − p)·FPR)

Paydayla çarp: PPV·[p·TPR + (1 − p)·FPR] = p·TPR ⇒ (1 − p)·FPR·PPV = p·TPR·(1 − PPV)

Düzenle: **FPR = p / (1 − p) · (1 − PPV) / PPV · TPR**

A: p = 0,5 ⇒ 0,5/0,5 · 0,2/0,8 · 0,8 = 1 · 0,25 · 0,8 = **0,20** ✓

Özdeşliği A grubuyla doğrulayalım.

Ölçütler Neden Birlikte Sağlanamaz?

p = taban oranı, n kişi ⇒ TP = p·n·TPR, FP = (1 − p)·n·FPR

PPV = TP / (TP + FP) = p·TPR / (p·TPR + (1 − p)·FPR)

Paydayla çarp: PPV·[p·TPR + (1 − p)·FPR] = p·TPR ⇒ (1 − p)·FPR·PPV = p·TPR·(1 − PPV)

Düzenle: **FPR = p / (1 − p) · (1 − PPV) / PPV · TPR**

A: p = 0,5 ⇒ 0,5/0,5 · 0,2/0,8 · 0,8 = 1 · 0,25 · 0,8 = **0,20** ✓

B: p = 0,32 ⇒ 0,32/0,68 · 0,25 · 0,5 = **0,059** ✓

B grubuyla doğrulayalım: tablodaki 4/68 ile aynı.

Ölçütler Neden Birlikte Sağlanamaz?

p = taban oranı, n kişi ⇒ TP = p·n·TPR, FP = (1 − p)·n·FPR

PPV = TP / (TP + FP) = p·TPR / (p·TPR + (1 − p)·FPR)

Paydayla çarp: PPV·[p·TPR + (1 − p)·FPR] = p·TPR ⇒ (1 − p)·FPR·PPV = p·TPR·(1 − PPV)

Düzenle: **FPR = p / (1 − p) · (1 − PPV) / PPV · TPR**

A: p = 0,5 ⇒ 0,5/0,5 · 0,2/0,8 · 0,8 = 1 · 0,25 · 0,8 = **0,20** ✓

B: p = 0,32 ⇒ 0,32/0,68 · 0,25 · 0,5 = **0,059** ✓

PPV eşit (0,8) ama p farklı ⇒ p/(1 − p) çarpanı farklı ⇒ TPR ve FPR **ikisi birden** eşit olamaz.

Sonuç: taban oranları farklıysa tahmin eşitliği ve eşitlenmiş olasılıklar birlikte sağlanamaz (mükemmel tahmin hariç).

Eşitlenmiş Olasılıkları Zorlarsak Ne Olur?

Hedef: B'de de TPR = 0,8 ve FPR = 0,2 (A ile aynı). B: 32 öder, 68 ödemez.

B için bu hata oranlarını veren, daha iyi ayırt eden (varsayımsal) bir model düşünelim. Bugünkü model (yalnız 0,8 ve 0,2 puanı) eşiği değiştirerek ya da rastgele onay vererek bu hata oranlarına ulaşamaz.

Eşitlenmiş Olasılıkları Zorlarsak Ne Olur?

Hedef: B'de de TPR = 0,8 ve FPR = 0,2 (A ile aynı). B: 32 öder, 68 ödemez.

TP = 32 × 0,8 = 25,6, FP = 68 × 0,2 = 13,6 (beklenen sayılar)

Hata oranları sabitlenince karışıklık matrisinin üst ve alt satırları belirlenir.

Eşitlenmiş Olasılıkları Zorlarsak Ne Olur?

Hedef: B'de de TPR = 0,8 ve FPR = 0,2 (A ile aynı). B: 32 öder, 68 ödemez.

TP = 32 × 0,8 = 25,6, FP = 68 × 0,2 = 13,6 (beklenen sayılar)

Onaylanan = 25,6 + 13,6 = 39,2 ⇒ seçim oranı 0,39

B'de onay oranı 0,2'den 0,39'a çıkar.

Eşitlenmiş Olasılıkları Zorlarsak Ne Olur?

Hedef: B'de de TPR = 0,8 ve FPR = 0,2 (A ile aynı). B: 32 öder, 68 ödemez.

TP = 32 × 0,8 = 25,6, FP = 68 × 0,2 = 13,6 (beklenen sayılar)

Onaylanan = 25,6 + 13,6 = 39,2 ⇒ seçim oranı 0,39

PPV_B = 25,6 / 39,2 = **0,65** < PPV_A = 0,80

Onaylananların ödeme oranı düşer.

Eşitlenmiş Olasılıkları Zorlarsak Ne Olur?

Hedef: B'de de TPR = 0,8 ve FPR = 0,2 (A ile aynı). B: 32 öder, 68 ödemez.

TP = 32 × 0,8 = 25,6, FP = 68 × 0,2 = 13,6 (beklenen sayılar)

Onaylanan = 25,6 + 13,6 = 39,2 ⇒ seçim oranı 0,39

PPV_B = 25,6 / 39,2 = **0,65** < PPV_A = 0,80

Eşitlenmiş olasılıklar sağlandı, ama tahmin eşitliği bozuldu: B'deki "onay" artık daha düşük bir ödeme olasılığı anlatıyor (model yalnız onay/ret verdiği için bozulan ölçüt PPV'dir).

Hangi ölçüt korunmalı? Matematik değil, bağlam (zararın türü, hukuk) karar verir.

Kalibrasyon ve Denge: Kleinberg Teoremi

Pozitif sınıf dengesi: ödeyenlerin ortalama puanı gruplar arasında eşit olmalı.

Puan düzeyinde eşitlenmiş olasılıkların karşılığı: sınıf dengesi (balance).

Kalibrasyon ve Denge: Kleinberg Teoremi

Pozitif sınıf dengesi: ödeyenlerin ortalama puanı gruplar arasında eşit olmalı.

A, Y=1: (40·0,8 + 10·0,2)/50 = **0,68**; B, Y=1: (16·0,8 + 16·0,2)/32 = **0,50**

Kalibre modelimizde B'nin ödeyen üyeleri ortalamada daha düşük puan alıyor.

Kalibrasyon ve Denge: Kleinberg Teoremi

Pozitif sınıf dengesi: ödeyenlerin ortalama puanı gruplar arasında eşit olmalı.

A, Y=1: (40·0,8 + 10·0,2)/50 = **0,68**; B, Y=1: (16·0,8 + 16·0,2)/32 = **0,50**

A, Y=0: (10·0,8 + 40·0,2)/50 = **0,32**; B, Y=0: (4·0,8 + 64·0,2)/68 = **0,235**

Ödemeyenlerde de denge yok. Kalibrasyon sağlandığı hâlde denge bozuk.

Kalibrasyon ve Denge: Kleinberg Teoremi

Pozitif sınıf dengesi: ödeyenlerin ortalama puanı gruplar arasında eşit olmalı.

A, Y=1: (40·0,8 + 10·0,2)/50 = **0,68**; B, Y=1: (16·0,8 + 16·0,2)/32 = **0,50**

A, Y=0: (10·0,8 + 40·0,2)/50 = **0,32**; B, Y=0: (4·0,8 + 64·0,2)/68 = **0,235**

Teorem (Kleinberg, Mullainathan, Raghavan 2017): kalibrasyon ve iki sınıf dengesi ancak mükemmel tahminde *ya da* eşit taban oranlarında birlikte sağlanabilir.

Örneğimizde taban oranları farklı (0,5 ve 0,32) ve tahmin mükemmel değil: ölçütlerin çatışması kaçınılmazdır.

Varsayımı Değiştir: Ya Adalet Ölçütünü Değiştirirsek?

grup	onay: öder / ödemez	ret: öder / ödemez	taban oranı
A	40 / 10	10 / 40	0,50
B	16 / 4	16 / 64	0,32

Her grupta 100 başvuran, aynı model, aynı eşik (0,5). **Bu model adil mi?**

Varsayımı Değiştir: Ya Adalet Ölçütünü Değiştirirsek?

grup	onay: öder / ödemez	ret: öder / ödemez	taban oranı
A	40 / 10	10 / 40	0,50
B	16 / 4	16 / 64	0,32

Her grupta 100 başvuran, aynı model, aynı eşik (0,5). **Bu model adil mi?**

Ölçüt	A	B	Sağlanıyor mu?
Demografik eşitlik (onay oranı)	0,50	0,20	**hayır**
Eşitlenmiş olasılıklar (TPR; FPR)	0,80; 0,20	0,50; 0,059	**hayır**
Tahmin eşitliği (PPV)	0,80	0,80	**evet**

Varsayımı Değiştir: Ya Adalet Ölçütünü Değiştirirsek?

grup	onay: öder / ödemez	ret: öder / ödemez	taban oranı
A	40 / 10	10 / 40	0,50
B	16 / 4	16 / 64	0,32

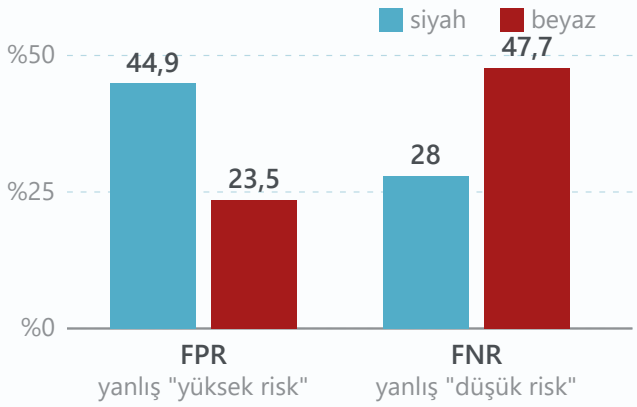
Her grupta 100 başvuran, aynı model, aynı eşik (0,5). **Bu model adil mi?**

Ölçüt	A	B	Sağlanıyor mu?
Demografik eşitlik (onay oranı)	0,50	0,20	**hayır**
Eşitlenmiş olasılıklar (TPR; FPR)	0,80; 0,20	0,50; 0,059	**hayır**
Tahmin eşitliği (PPV)	0,80	0,80	**evet**

Cevap ölçüte bağlı. Kurumun ölçütüyle adil, başvuranın ölçütleriyle adil değil. Taban oranları farklıysa (0,50 ve 0,32) bu ölçütler, tahmin mükemmel olmadıkça, aynı anda sağlanamaz (Chouldechova, 2017; Kleinberg vd., 2017). Hangi ölçütün seçileceği bir değer kararıdır.

ÖLÇÜTLERİN UYUMSUZLUĞU

Gerçek Bir Örnek: COMPAS Tartışması



ProPublica (2016), Broward County verisi: yüzde oranlar

ProPublica

Yeniden suç işlemeyen siyah sanıkların %44,9'u "yüksek riskli" etiketlenmiş, beyazlarda %23,5: **FPR eşit değil**. FNR (yanlış negatif oranı: yeniden suç işleyenlerden "düşük risk" denenlerin payı) için tablo tersine döner.

Northpointe (şirket)

Aynı risk puanı iki grupta benzer yeniden suç oranına karşılık geliyor: **kalibrasyon / tahmin eşitliği var**.

İki taraf da hesabında haklıydı. Taban oranları farklı olduğundan iki ölçüt birlikte sağlanamazdı (Chouldechova özdeşliği): tartışma aslında **hangi adalet tanımının** seçileceğiyle ilgilidir. Tek bir doğruluk puanı yerine grup bazlı hata oranları raporlanmalıdır.

ÖLÇÜTLERİN UYUMSUZLUĞU

Adalet ve Doğruluk Ödünleşimi

- Model doğruluğu tek başına hataların **kimler üzerinde yoğunlaştığını** göstermez: örneğimizde doğruluk iki grupta da %80, ama TPR 0,8'e 0,5.
- Bir adalet kısıtı toplam hatayı artırabilir; asıl seçim **hangi hataların kabul edilebilir** olduğudur (yanlış ret mi, yanlış onay mı?).

Ön işleme

Veriyi yeniden ağırlıklandırmak, temsil eksikliğini gidermek.

Eğitim sırasında

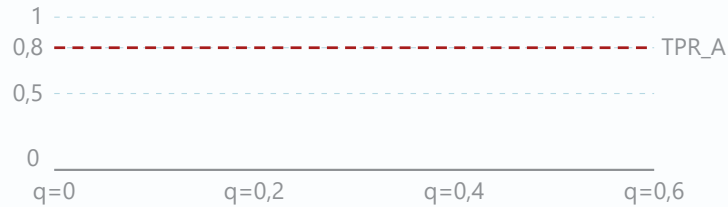
Eğitimde en aza indirilen hata ölçüsüne (kayıp fonksiyonu) adalet kısıtı eklemek (bir miktar doğruluktan vazgeçmek).

Son işleme

Gruba özgü eşikler (Hardt vd.): hukuken tartışmalı olabilir.

Kullanım bağlamı, hukuki gereklilikler, düzenlemeler ve etkilenen grupların görüşleri ölçüt seçimine yön vermelidir. Örnek: düşük gelirli başvuranları adil değerlendiren kredi skorlaması.

Kod İzleme: Rastgeleleştirme ile Fırsat Eşitliği



q = 0
TP = 16 FP = 4

q = 0: yalnız 0,8 puanlılar onaylanır (bugünkü model). TP = 16, FP = 4.

Adım 1 / 9

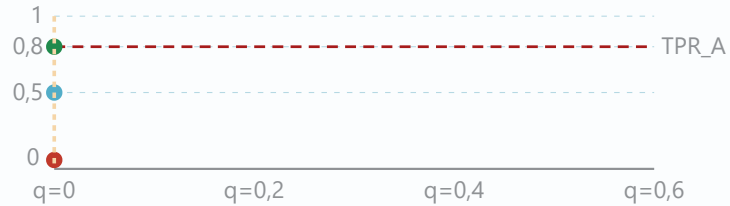
```
1  def b_grubu(q):   # 0,2 puanlıyı q ile onayla
▶     TP = 16 + 16 * q
▶     FP = 4 + 64 * q
4     tpr = TP / 32
5     fpr = FP / 68
6     dogru = TP + (68 - FP)   # TP + TN
7     return tpr, fpr, round(dogru / 100, 3)
8
▶  for q in [0, 0.2, 0.4, 0.6]:
▶     tpr, fpr, d = b_grubu(q)
11    print(q, round(tpr, 2), round(fpr, 2), d)
```

DEĞİŞKENLER

`q = 0` `TP = 16` `FP = 4`

ÇIKTI

Kod İzleme: Rastgeleleştirme ile Fırsat Eşitliği



q = 0
TP = 16 FP = 4
TPR_B = 0,5
FPR_B = 0,06
doğruluk_B = 0,8

TPR = 16/32 = 0,5; FPR = 4/68 ≈ 0,06; doğruluk (16 + 64)/100 = 0,80.

Adım 2 / 9

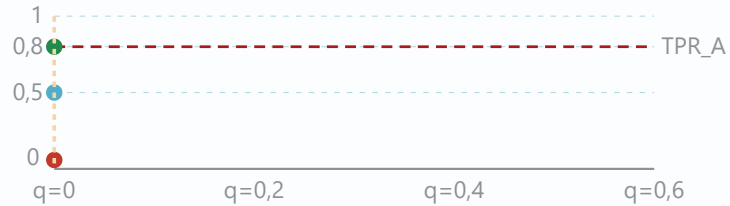
```
1  def b_grubu(q):   # 0,2 puanlıyı q ile onayla
2    TP = 16 + 16 * q
3    FP = 4 + 64 * q
▶    tpr = TP / 32
▶    fpr = FP / 68
▶    dogru = TP + (68 - FP)   # TP + TN
▶    return tpr, fpr, round(dogru / 100, 3)
8
9  for q in [0, 0.2, 0.4, 0.6]:
10   tpr, fpr, d = b_grubu(q)
11   print(q, round(tpr, 2), round(fpr, 2), d)
```

DEĞİŞKENLER

`q = 0` `tpr = 0.5` `fpr = 0.059` `dogru = 80`

ÇIKTI

Kod İzleme: Rastgeleleştirme ile Fırsat Eşitliği



q = 0,2
TP = 19,2 FP = 16,8

q = 0,2: 0,2 puanlı 80 kişinin %20'si onaylanır. TP = 16 + 16·q = 19,2, FP = 4 + 64·q = 16,8.

Adım 3 / 9

```
1  def b_grubu(q):   # 0,2 puanlıyı q ile onayla
2      TP = 16 + 16 * q
3      FP = 4 + 64 * q
4      tpr = TP / 32
5      fpr = FP / 68
6      dogru = TP + (68 - FP)   # TP + TN
7      return tpr, fpr, round(dogru / 100, 3)
8
9  for q in [0, 0.2, 0.4, 0.6]:
10     tpr, fpr, d = b_grubu(q)
11     print(q, round(tpr, 2), round(fpr, 2), d)
```

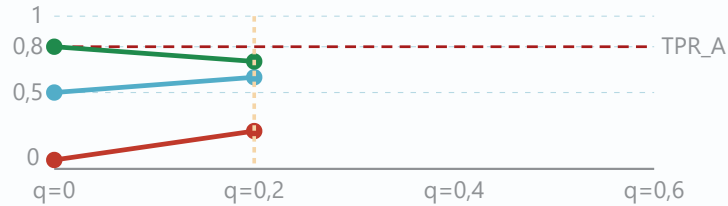
DEĞİŞKENLER

q = 0.2 | TP = 19.2 | FP = 16.8

ÇIKTI

```
0 0.5 0.06 0.8
```

Kod İzleme: Rastgeleleştirme ile Fırsat Eşitliği



q = 0,2
TP = 19,2 FP = 16,8
TPR_B = 0,6
FPR_B = 0,25
doğruluk_B = 0,704

TPR 0,6'ya çıktı ama doğruluk 0,704'e düştü: yeni onaylanan her 5 kişiden 4'ü ödemeyecek olanlardan.

Adım 4 / 9

```
1  def b_grubu(q):   # 0,2 puanlıyı q ile onayla
2    TP = 16 + 16 * q
3    FP = 4 + 64 * q
▶    tpr = TP / 32
▶    fpr = FP / 68
▶    dogru = TP + (68 - FP)   # TP + TN
▶    return tpr, fpr, round(dogru / 100, 3)
8
9  for q in [0, 0.2, 0.4, 0.6]:
10   tpr, fpr, d = b_grubu(q)
11   print(q, round(tpr, 2), round(fpr, 2), d)
```

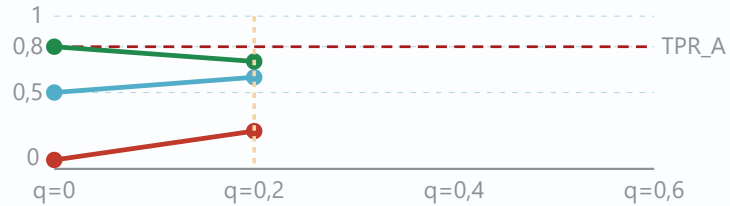
DEĞİŞKENLER

`q = 0.2` `tpr = 0.6` `fpr = 0.247` `dogru = 70.4`

ÇIKTI

```
0 0.5 0.06 0.8
```

Kod İzleme: Rastgeleleştirme ile Fırsat Eşitliği



q = 0,4
TP = 22,4 FP = 29,6

q = 0,4: 0,2 puanlı 80 kişinin %40'ı onaylanır. TP = 16 + 16·q = 22,4, FP = 4 + 64·q = 29,6.

Adım 5 / 9

```
1  def b_grubu(q):   # 0,2 puanlıyı q ile onayla
▶     TP = 16 + 16 * q
▶     FP = 4 + 64 * q
4     tpr = TP / 32
5     fpr = FP / 68
6     dogru = TP + (68 - FP)   # TP + TN
7     return tpr, fpr, round(dogru / 100, 3)
8
▶  for q in [0, 0.2, 0.4, 0.6]:
▶     tpr, fpr, d = b_grubu(q)
11    print(q, round(tpr, 2), round(fpr, 2), d)
```

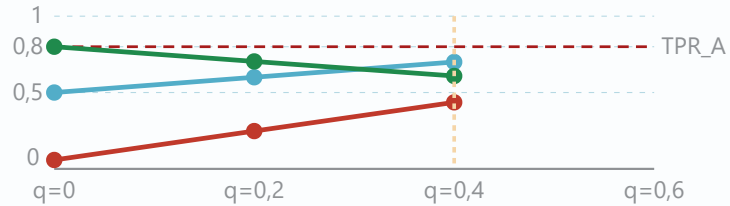
DEĞİŞKENLER

`q = 0.4` `TP = 22.4` `FP = 29.6`

ÇIKTI

```
0 0.5 0.06 0.8
0.2 0.6 0.25 0.704
```

Kod İzleme: Rastgeleleştirme ile Fırsat Eşitliği



q = 0,4
TP = 22,4 FP = 29,6
TPR_B = 0,7
FPR_B = 0,44
doğruluk_B = 0,608

TPR 0,7; FPR 0,44'e fırladı, doğruluk 0,608.


```
1  def b_grubu(q):   # 0,2 puanlıyı q ile onayla
2    TP = 16 + 16 * q
3    FP = 4 + 64 * q
▶    tpr = TP / 32
▶    fpr = FP / 68
▶    dogru = TP + (68 - FP)   # TP + TN
▶    return tpr, fpr, round(dogru / 100, 3)
8
9  for q in [0, 0.2, 0.4, 0.6]:
10   tpr, fpr, d = b_grubu(q)
11   print(q, round(tpr, 2), round(fpr, 2), d)
```

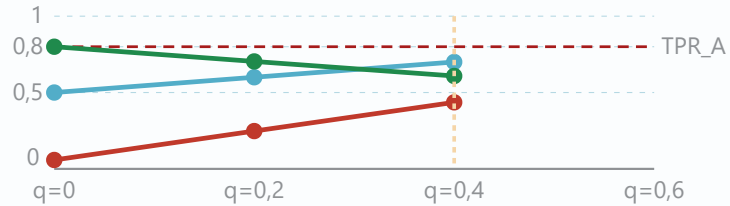
DEĞİŞKENLER

`q = 0.4` `tpr = 0.7` `fpr = 0.435` `dogru = 60.8`

ÇIKTI

```
0 0.5 0.06 0.8
0.2 0.6 0.25 0.704
```

Kod İzleme: Rastgeleleştirme ile Fırsat Eşitliği



q = 0,6
TP = 25,6 FP = 42,4

q = 0,6: 0,2 puanlı 80 kişinin %60'ı onaylanır. TP = 16 + 16·q = 25,6, FP = 4 + 64·q = 42,4.

Adım 7 / 9

```
1  def b_grubu(q):   # 0,2 puanlıyı q ile onayla
2      TP = 16 + 16 * q
3      FP = 4 + 64 * q
4      tpr = TP / 32
5      fpr = FP / 68
6      dogru = TP + (68 - FP)   # TP + TN
7      return tpr, fpr, round(dogru / 100, 3)
8
9  for q in [0, 0.2, 0.4, 0.6]:
10     tpr, fpr, d = b_grubu(q)
11     print(q, round(tpr, 2), round(fpr, 2), d)
```

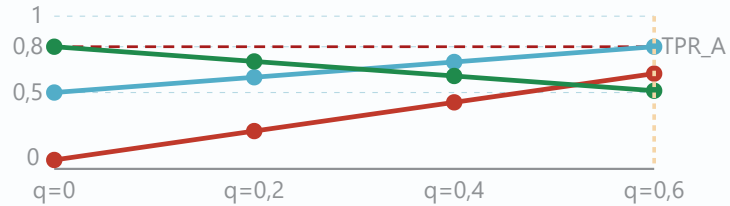
DEĞİŞKENLER

q = 0.6 | TP = 25.6 | FP = 42.4

ÇIKTI

```
0 0.5 0.06 0.8
0.2 0.6 0.25 0.704
0.4 0.7 0.44 0.608
```

Kod İzleme: Rastgeleleştirme ile Fırsat Eşitliği



q = 0,6
TP = 25,6 FP = 42,4
TPR_B = 0,8
FPR_B = 0,62
doğruluk_B = 0,512

TPR_B = 0,8 = TPR_A: **fırsat eşitliği sağlandı**. Bedeli: doğruluk 0,512, FPR 0,62.

Adım 8 / 9

```
1  def b_grubu(q):   # 0,2 puanlıyı q ile onayla
2    TP = 16 + 16 * q
3    FP = 4 + 64 * q
▶    tpr = TP / 32
▶    fpr = FP / 68
▶    dogru = TP + (68 - FP)   # TP + TN
▶    return tpr, fpr, round(dogru / 100, 3)
8
9  for q in [0, 0.2, 0.4, 0.6]:
10   tpr, fpr, d = b_grubu(q)
11   print(q, round(tpr, 2), round(fpr, 2), d)
```

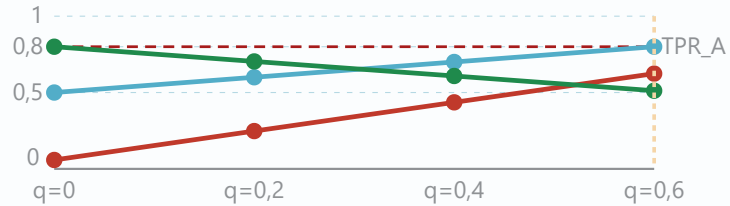
DEĞİŞKENLER

`q = 0.6` `tpr = 0.8` `fpr = 0.624` `dogru = 51.2`

ÇIKTI

```
0 0.5 0.06 0.8
0.2 0.6 0.25 0.704
0.4 0.7 0.44 0.608
```

Kod İzleme: Rastgeleleştirme ile Fırsat Eşitliği



q = 0,6
TP = 25,6 FP = 42,4
TPR_B = 0,8
FPR_B = 0,62
doğruluk_B = 0,512

Fırsat eşitliği için 0,2 puanlıların %60'ı rastgele onaylandı: ödemeyecek 64 kişiden ≈ 38'i kredi alır. FPR, A'dakinin (0,2) üç katı: eşitlenmiş olasılıklar yine yok.

Adım 9 / 9

```
1  def b_grubu(q):   # 0,2 puanlıyı q ile onayla
2    TP = 16 + 16 * q
3    FP = 4 + 64 * q
4    tpr = TP / 32
5    fpr = FP / 68
6    dogru = TP + (68 - FP)   # TP + TN
7    return tpr, fpr, round(dogru / 100, 3)
8
9  for q in [0, 0.2, 0.4, 0.6]:
10   tpr, fpr, d = b_grubu(q)
▶    print(q, round(tpr, 2), round(fpr, 2), d)
```

DEĞİŞKENLER

`q = 0.6` `tpr = 0.8` `d = 0.512`

ÇIKTI

```
0 0.5 0.06 0.8
0.2 0.6 0.25 0.704
0.4 0.7 0.44 0.608
0.6 0.8 0.62 0.512
```

Adil Bölüşüm: Pasta Kesme (Cake Cutting)

- Bölünebilir bir kaynak (pasta = [0, 1] aralığı: zaman, arazi, işlem süresi) n kişi arasında paylaştırılacak.
- Her kişinin kendi **değer fonksiyonu** v_i var: toplam pasta 1 değerinde, ama kişiler farklı kısımlara farklı değer verir.
- Algoritma kişilere yalnızca soru sorar (Robertson–Webb modeli): "şu parçaya ne değer verirsin?" ve "değeri belli bir miktar olan parça nerede biter?". X_i: i. kişinin aldığı parça; $v_i(X_i)$: bu parçanın i. kişinin gözündeki değeri.

Orantılılık (proportionality)

Herkes kendi gözünde en az **1/n** alır: $v_i(X_i) \geq 1/n$.

Kıskançsızlık (envy-freeness)

Kimse başkasının parçasını kendininkine tercih etmez: $v_i(X_i) \geq v_i(X_j)$.

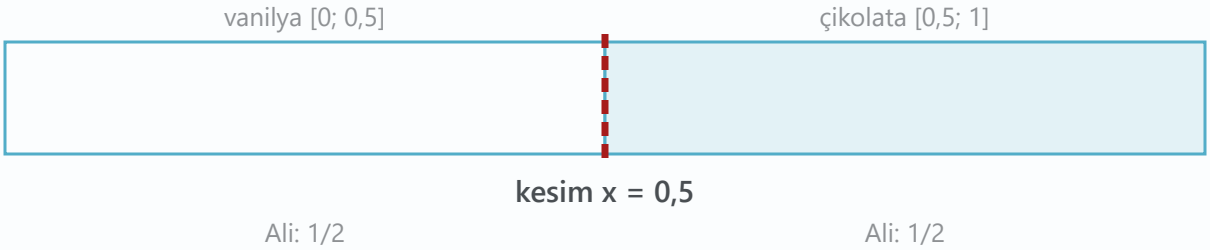
Oyun kuramı bakışı: kişiler stratejik olabilir. İyi bir protokol, dürüst davranan herkese **garanti** verir; başkaları ne yaparsa yapsın.

Kes-ve-Seç (Cut and Choose) Adım Adım



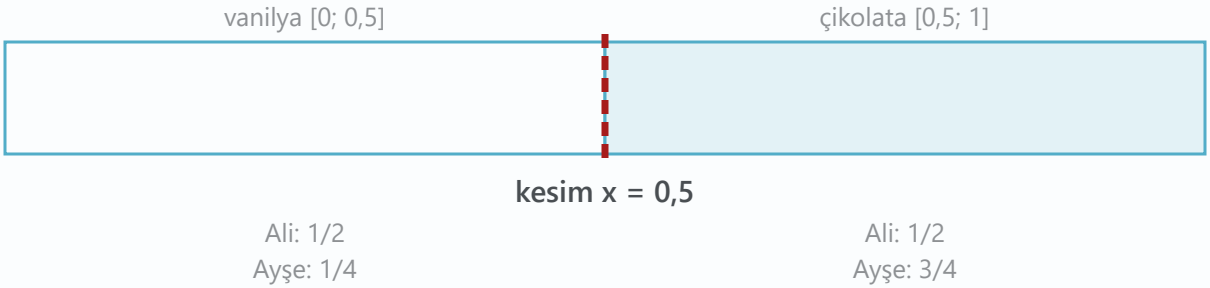
Ali pastanın her yerine eşit değer verir. Ayşe çikolatayı vanilyadan **3 kat** daha çok sever: $v_{Ayşe}$(vanilya) = 1/4, $v_{Ayşe}$(çikolata) = 3/4.

Kes-ve-Seç (Cut and Choose) Adım Adım



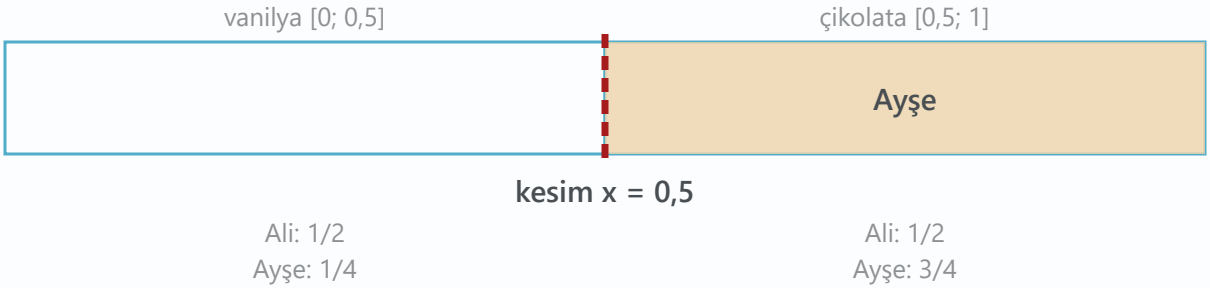
1. Ali keser: kendi gözünde iki eşit parça yapar ⇒ x = 1/2, her parça Ali için 1/2.

Kes-ve-Seç (Cut and Choose) Adım Adım



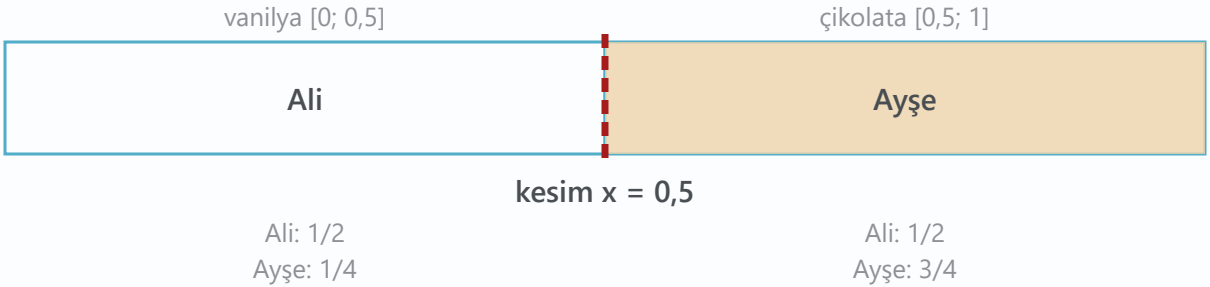
2. Ayşe değerlendirir: sol parça 1/4, sağ parça 3/4.

Kes-ve-Seç (Cut and Choose) Adım Adım



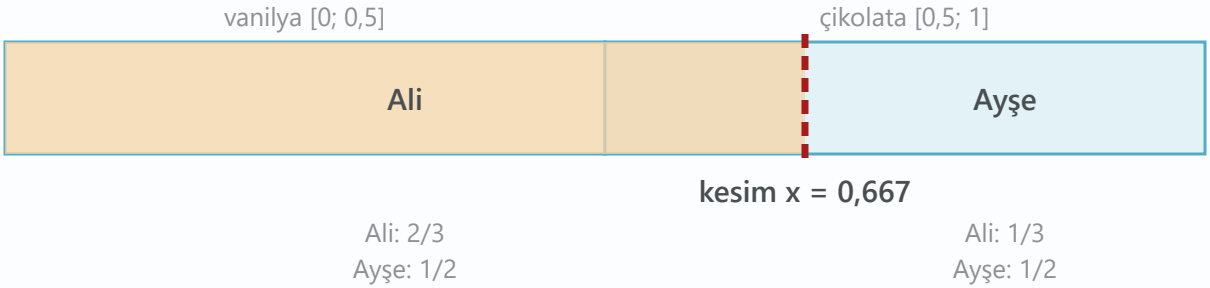
3. Ayşe seçer: değeri büyük olan sağ parçayı (çikolata) alır.

Kes-ve-Seç (Cut and Choose) Adım Adım



Sonuç: Ali 1/2 ≥ 1/2, Ayşe 3/4 ≥ 1/2 ⇒ **orantılı**; Ali 1/2 = 1/2, Ayşe 3/4 > 1/4 ⇒ **kıskançsız**.

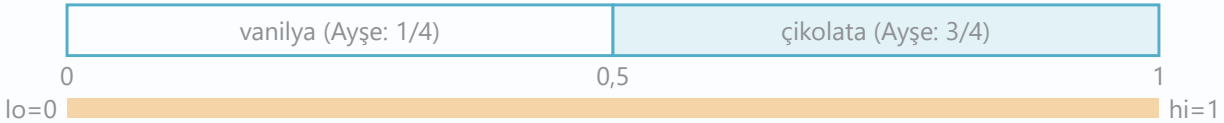
Kes-ve-Seç (Cut and Choose) Adım Adım



Roller değişirse: Ayşe x = 2/3'ten keser (kendi gözünde 1/2 – 1/2). Ali sol parçayı seçer ve 2/3 alır; Ayşe 1/2. Seçen, kesenden avantajlıdır.

ADİL BÖLÜŞÜM

Kod İzleme: Kesim Sorgusu İkiye Bölme ile



Robertson–Webb *kesim sorgusu*: "[0, x] parçası senin için 1/2 olsun, x nerede?" Ayşe'nin değer fonksiyonu artan olduğundan ikiye bölme (bisection) ile bulunur.

Adım 1 / 10

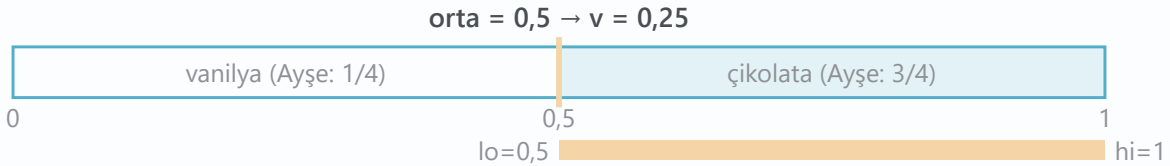
```
1  def v_ayse(x):          # [0, x] parçasının değeri
2    if x <= 0.5: return x / 2
3    return 0.25 + 1.5 * (x - 0.5)
4  def kes_noktasi(v, hedef, adim):
▶    lo, hi = 0.0, 1.0
6    for _ in range(adim):
7      orta = (lo + hi) / 2
8      if v(orta) < hedef: lo = orta
9      else: hi = orta
10   return (lo + hi) / 2
▶  print(round(kes_noktasi(v_ayse, 0.5, 8), 3))
```

DEĞİŞKENLER

`hedef = 0.5` `lo = 0.0` `hi = 1.0`

ÇIKTI

Kod İzleme: Kesim Sorgusu İkiye Bölme ile



1. sorgu: orta = 0,5: vanilyanın tamamı Ayşe için 1/4 = 0,25 < 0,5 ⇒ lo = 0,5. Aralık yarıya iner (genişlik 1/2).

Adım 2 / 10

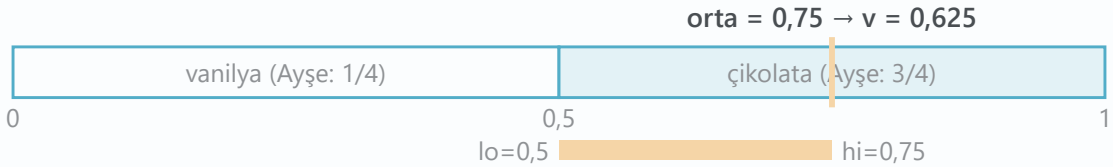
```
1  def v_ayse(x):          # [0, x] parçasının değeri
▶    if x <= 0.5: return x / 2
3    return 0.25 + 1.5 * (x - 0.5)
4  def kes_noktasi(v, hedef, adim):
5    lo, hi = 0.0, 1.0
▶    for _ in range(adim):
▶      orta = (lo + hi) / 2
▶      if v(orta) < hedef: lo = orta
9      else: hi = orta
10   return (lo + hi) / 2
11 print(round(kes_noktasi(v_ayse, 0.5, 8), 3))
```

DEĞİŞKENLER

`_ = 0` `orta = 0.5` `v(orta) = 0.25` `lo = 0.5` `hi = 1.0`

ÇIKTI

Kod İzleme: Kesim Sorgusu İkiye Bölme ile



2. sorgu: v(0,75) = 0,625 ≥ 0,5 ⇒ kesim solda ya da burada: hi = orta. Aralık yarıya iner (genişlik 1/4).

Adım 3 / 10

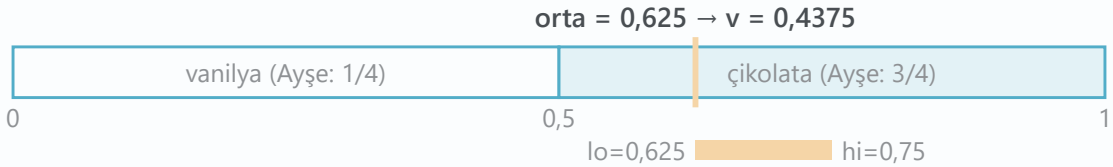
```
1  def v_ayse(x):          # [0, x] parçasının değeri
▶    if x <= 0.5: return x / 2
▶    return 0.25 + 1.5 * (x - 0.5)
4  def kes_noktasi(v, hedef, adim):
5    lo, hi = 0.0, 1.0
▶    for _ in range(adim):
▶      orta = (lo + hi) / 2
8      if v(orta) < hedef: lo = orta
▶      else: hi = orta
10   return (lo + hi) / 2
11 print(round(kes_noktasi(v_ayse, 0.5, 8), 3))
```

DEĞİŞKENLER

`_ = 1` `orta = 0.75` `v(orta) = 0.625` `lo = 0.5` `hi = 0.75`

ÇIKTI

Kod İzleme: Kesim Sorgusu İkiye Bölme ile



3. sorgu: v(0,625) = 0,4375 < 0,5 ⇒ kesim daha sağda: lo = orta. Aralık yarıya iner (genişlik 1/8).

Adım 4 / 10

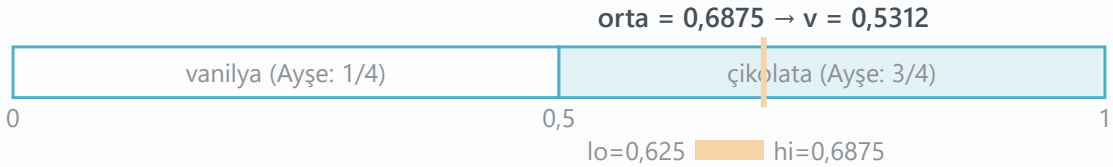
```
1  def v_ayse(x):          # [0, x] parçasının değeri
▶    if x <= 0.5: return x / 2
▶    return 0.25 + 1.5 * (x - 0.5)
4  def kes_noktasi(v, hedef, adim):
5    lo, hi = 0.0, 1.0
▶    for _ in range(adim):
▶      orta = (lo + hi) / 2
▶      if v(orta) < hedef: lo = orta
9      else: hi = orta
10   return (lo + hi) / 2
11 print(round(kes_noktasi(v_ayse, 0.5, 8), 3))
```

DEĞİŞKENLER

`_ = 2` `orta = 0.625` `v(orta) = 0.4375` `lo = 0.625` `hi = 0.75`

ÇIKTI

Kod İzleme: Kesim Sorgusu İkiye Bölme ile



4. sorgu: v(0,6875) = 0,5312 ≥ 0,5 ⇒ kesim solda ya da burada: hi = orta. Aralık yarıya iner (genişlik 1/16).

Adım 5 / 10

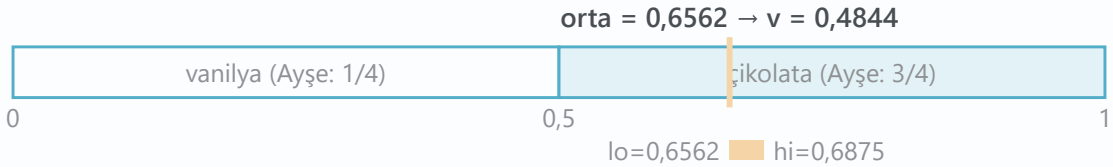
```
1  def v_ayse(x):          # [0, x] parçasının değeri
▶    if x <= 0.5: return x / 2
▶    return 0.25 + 1.5 * (x - 0.5)
4  def kes_noktasi(v, hedef, adim):
5    lo, hi = 0.0, 1.0
▶    for _ in range(adim):
▶      orta = (lo + hi) / 2
8      if v(orta) < hedef: lo = orta
▶      else: hi = orta
10   return (lo + hi) / 2
11 print(round(kes_noktasi(v_ayse, 0.5, 8), 3))
```

DEĞİŞKENLER

`_ = 3` `orta = 0.6875` `v(orta) = 0.53125` `lo = 0.625` `hi = 0.6875`

ÇIKTI

Kod İzleme: Kesim Sorgusu İkiye Bölme ile



5. sorgu: v(0,6562) = 0,4844 < 0,5 ⇒ kesim daha sağda: lo = orta. Aralık yarıya iner (genişlik 1/32).

Adım 6 / 10

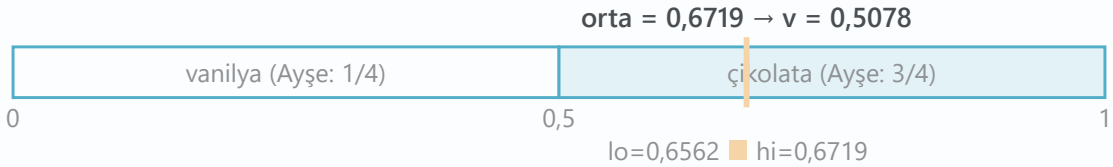
```
1  def v_ayse(x):          # [0, x] parçasının değeri
▶    if x <= 0.5: return x / 2
▶    return 0.25 + 1.5 * (x - 0.5)
4  def kes_noktasi(v, hedef, adim):
5    lo, hi = 0.0, 1.0
▶    for _ in range(adim):
▶      orta = (lo + hi) / 2
▶      if v(orta) < hedef: lo = orta
9      else: hi = orta
10   return (lo + hi) / 2
11 print(round(kes_noktasi(v_ayse, 0.5, 8), 3))
```

DEĞİŞKENLER

`_ = 4` `orta = 0.65625` `v(orta) = 0.48438` `lo = 0.65625` `hi = 0.6875`

ÇIKTI

Kod İzleme: Kesim Sorgusu İkiye Bölme ile



6. sorgu: v(0,6719) = 0,5078 ≥ 0,5 ⇒ kesim solda ya da burada: hi = orta. Aralık yarıya iner (genişlik 1/64).

Adım 7 / 10

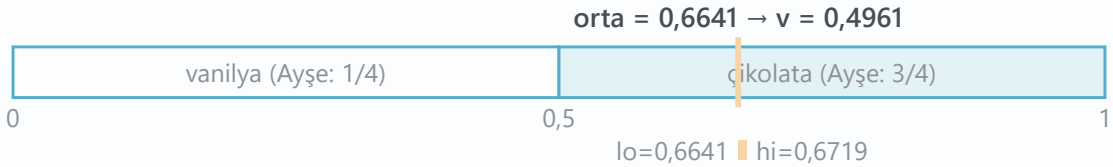
```
1  def v_ayse(x):          # [0, x] parçasının değeri
2    if x <= 0.5: return x / 2
3    return 0.25 + 1.5 * (x - 0.5)
4  def kes_noktasi(v, hedef, adim):
5    lo, hi = 0.0, 1.0
6    for _ in range(adim):
7      orta = (lo + hi) / 2
8      if v(orta) < hedef: lo = orta
9      else: hi = orta
10   return (lo + hi) / 2
11 print(round(kes_noktasi(v_ayse, 0.5, 8), 3))
```

DEĞİŞKENLER

`_ = 5` `orta = 0.671875` `v(orta) = 0.50781` `lo = 0.65625` `hi = 0.671875`

ÇIKTI

Kod İzleme: Kesim Sorgusu İkiye Bölme ile



7. sorgu: v(0,6641) = 0,4961 < 0,5 ⇒ kesim daha sağda: lo = orta. Aralık yarıya iner (genişlik 1/128).

Adım 8 / 10

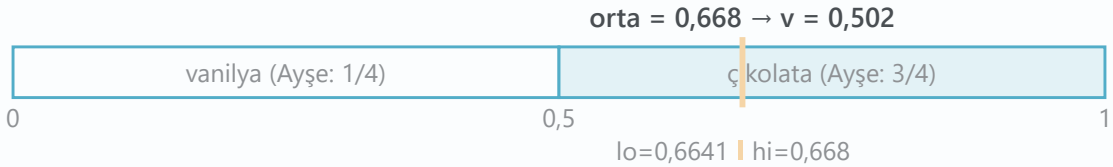
```
1  def v_ayse(x):          # [0, x] parçasının değeri
▶    if x <= 0.5: return x / 2
▶    return 0.25 + 1.5 * (x - 0.5)
4  def kes_noktasi(v, hedef, adim):
5    lo, hi = 0.0, 1.0
▶    for _ in range(adim):
▶      orta = (lo + hi) / 2
▶      if v(orta) < hedef: lo = orta
9      else: hi = orta
10   return (lo + hi) / 2
11 print(round(kes_noktasi(v_ayse, 0.5, 8), 3))
```

DEĞİŞKENLER

`_ = 6` `orta = 0.6640625` `v(orta) = 0.49609` `lo = 0.6640625` `hi = 0.671875`

ÇIKTI

Kod İzleme: Kesim Sorgusu İkiye Bölme ile



8. sorgu: v(0,668) = 0,502 ≥ 0,5 ⇒ kesim solda ya da burada: hi = orta. Aralık yarıya iner (genişlik 1/256).

Adım 9 / 10

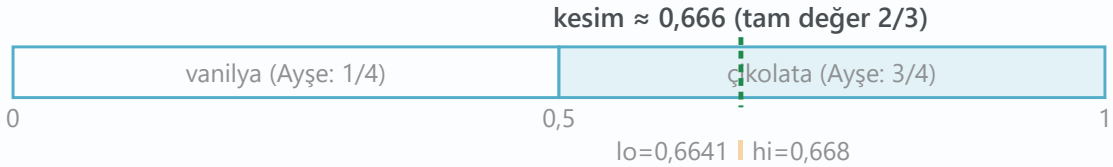
```
1  def v_ayse(x):          # [0, x] parçasının değeri
2    if x <= 0.5: return x / 2
3    return 0.25 + 1.5 * (x - 0.5)
4  def kes_noktasi(v, hedef, adim):
5    lo, hi = 0.0, 1.0
6    for _ in range(adim):
7      orta = (lo + hi) / 2
8      if v(orta) < hedef: lo = orta
9      else: hi = orta
10   return (lo + hi) / 2
11 print(round(kes_noktasi(v_ayse, 0.5, 8), 3))
```

DEĞİŞKENLER

`_ = 7` `orta = 0.66796875` `v(orta) = 0.50195` `lo = 0.6640625` `hi = 0.66796875`

ÇIKTI

Kod İzleme: Kesim Sorgusu İkiye Bölme ile



8 sorgudan sonra aralık 1/256 genişlikte; orta nokta 0,666. Gerçek kesim 2/3: Ayşe kesen olsaydı buradan keserdi.

Adım 10 / 10

```
1 def v_ayse(x):          # [0, x] parçasının değeri
2   if x <= 0.5: return x / 2
3   return 0.25 + 1.5 * (x - 0.5)
4 def kes_noktasi(v, hedef, adim):
5   lo, hi = 0.0, 1.0
6   for _ in range(adim):
7     orta = (lo + hi) / 2
8     if v(orta) < hedef: lo = orta
9     else: hi = orta
▶   return (lo + hi) / 2
▶ print(round(kes_noktasi(v_ayse, 0.5, 8), 3))
```

DEĞİŞKENLER

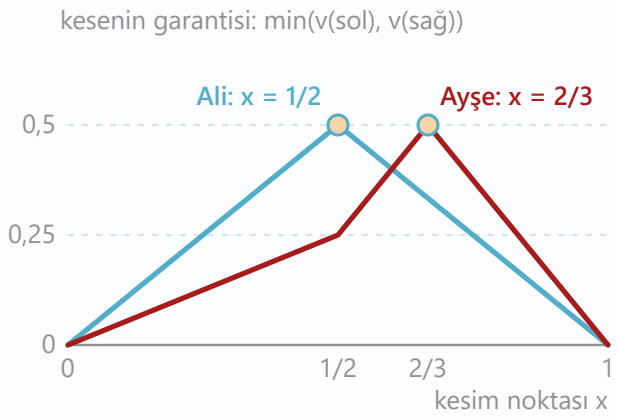
`lo = 0.6640625` `hi = 0.66796875`

`sonuç = 0.66602`

ÇIKTI

```
0.666
```

Kes-ve-Seç Neden Adil?



- **Kesen** hangi parçanın kalacağını bilmez; garantisi iki parçanın küçüğüdür. Bu garanti tam kendi gözündeki yarıda en büyüktür (1/2): dürüst kesmek **maksimin** stratejisidir.
- **Seçen** kendi gözünde büyük parçayı alır: değeri ≥ 1/2 ve ötekinden az değil, kesim nasıl olursa olsun.
- Kesen, seçenin tercihini bilirse kesimi ona göre kaydırıp 1/2'den fazlasını alabilir; ama *garantisi* 1/2'dir.

İki kişide orantılılık ⇔ kıskançsızlık: $v(X_1) \geq 1/2$ ise $v(X_2) = 1 - v(X_1) \leq 1/2 \leq v(X_1)$. Üç ve daha fazla kişide kıskançsızlık daha güçlüdür.

Orantılı Ama Kıskançlı: Üç Kişilik Örnek

kişi	X parçası	Y parçası	Z parçası	aldığı	≥ 1/3?
Ali	0,40	0,30	0,30		
Ayşe	0,20	0,35	0,45		
Cem	0,30	0,30	0,40		

Üç kişi, üç parça (X, Y, Z). Tablo her kişinin her parçaya verdiği değer; her satırın toplamı 1.

Adım 1 / 3

Son azalan (last diminisher)
n kişi için orantılı paylaşım.

Selfridge–Conway
3 kişi için kıskançsız, en fazla 5 kesim.

Aziz–Mackenzie (2016)
n kişi için kıskançsız; sonlu ama devasa sayıda soru.

Orantılı Ama Kıskançlı: Üç Kişilik Örnek

kişi	X parçası	Y parçası	Z parçası	aldığı	≥ 1/3?
Ali	**0,40**	0,30	0,30	X: 0,40	**evet**
Ayşe	0,20	**0,35**	0,45	Y: 0,35	**evet**
Cem	0,30	0,30	**0,40**	Z: 0,40	**evet**

Paylaşım: Ali → X, Ayşe → Y, Cem → Z. Herkes kendi gözünde en az 1/3 ≈ 0,33 alıyor ⇒ **orantılı**.

Son azalan (last diminisher)
n kişi için orantılı paylaşım.

Selfridge–Conway
3 kişi için kıskançsız, en fazla 5 kesim.

Aziz–Mackenzie (2016)
n kişi için kıskançsız; sonlu ama devasa sayıda soru.

Orantılı Ama Kıskançlı: Üç Kişilik Örnek

kişi	X parçası	Y parçası	Z parçası	aldığı	≥ 1/3?
Ali	**0,40**	0,30	0,30	X: 0,40	**evet**
Ayşe	0,20	**0,35**	**0,45**	Y: 0,35	**evet**
Cem	0,30	0,30	**0,40**	Z: 0,40	**evet**

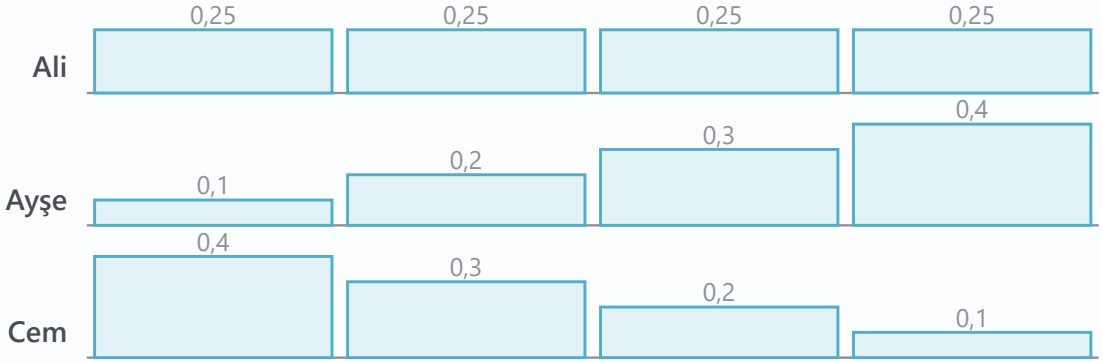
Ama Ayşe için Cem'in parçası Z (0,45) kendi parçası Y'den (0,35) değerli ⇒ Ayşe Cem'i **kıskanır**. Orantılı ama kıskançsız değil.

Son azalan (last diminisher)
n kişi için orantılı paylaşım.

Selfridge–Conway
3 kişi için kıskançsız, en fazla 5 kesim.

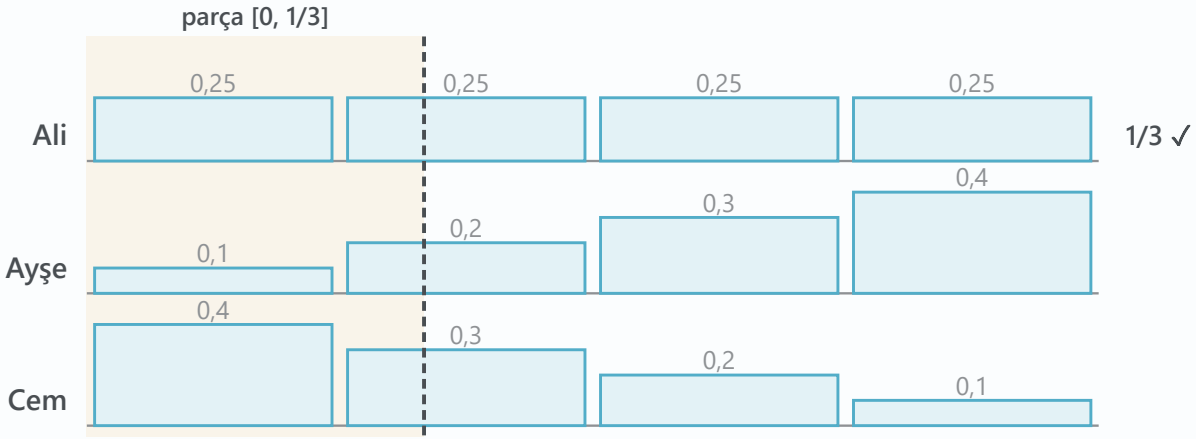
Aziz–Mackenzie (2016)
n kişi için kıskançsız; sonlu ama devasa sayıda soru.

Son Azalan (Last Diminisher) Adım Adım



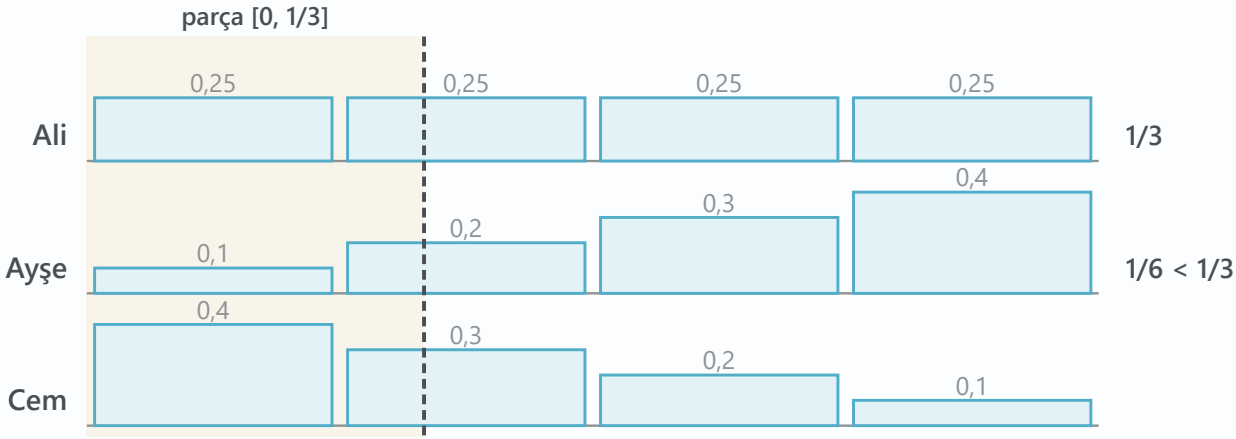
Üç kişi, pasta dört çeyrek. Çubuklar her kişinin çeyrek başına değeri (toplam 1). Ali tekdüze; Ayşe sağı, Cem solu seviyor.

Son Azalan (Last Diminisher) Adım Adım



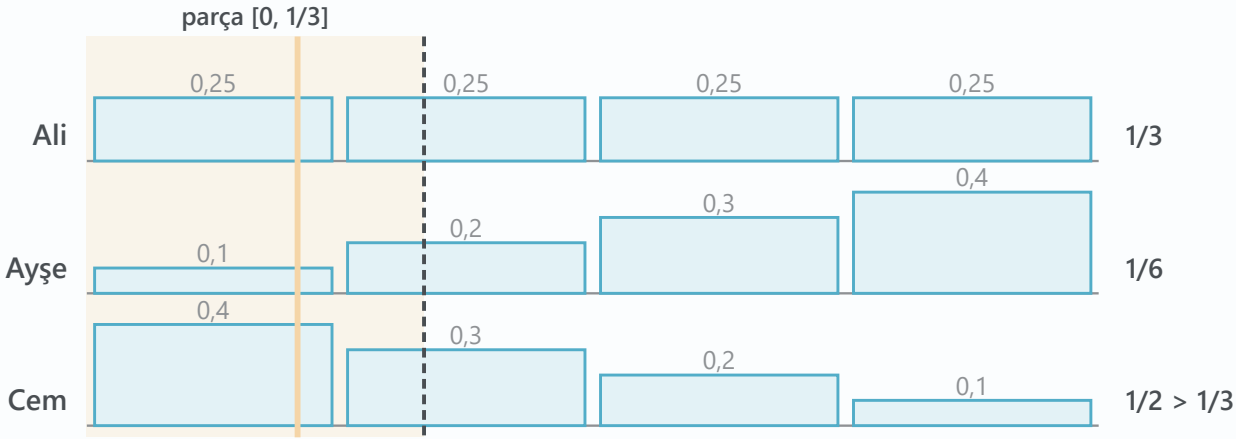
Ali kendi gözünde 1/3 değerinde bir parça keser: [0, 1/3].

Son Azalan (Last Diminisher) Adım Adım



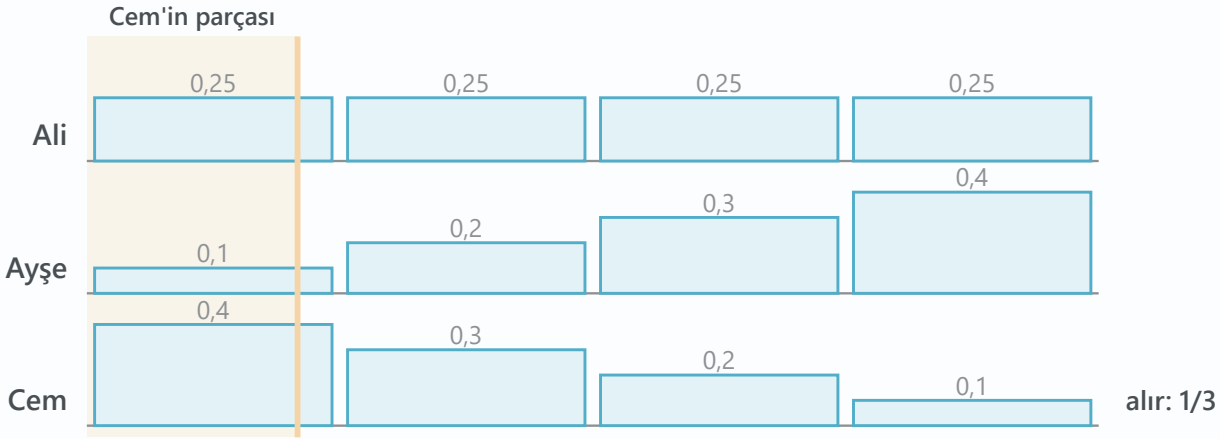
Ayşe parçaya 1/6 değer verir: 1/3'ten az, dokunmaz ("geçer").

Son Azalan (Last Diminisher) Adım Adım



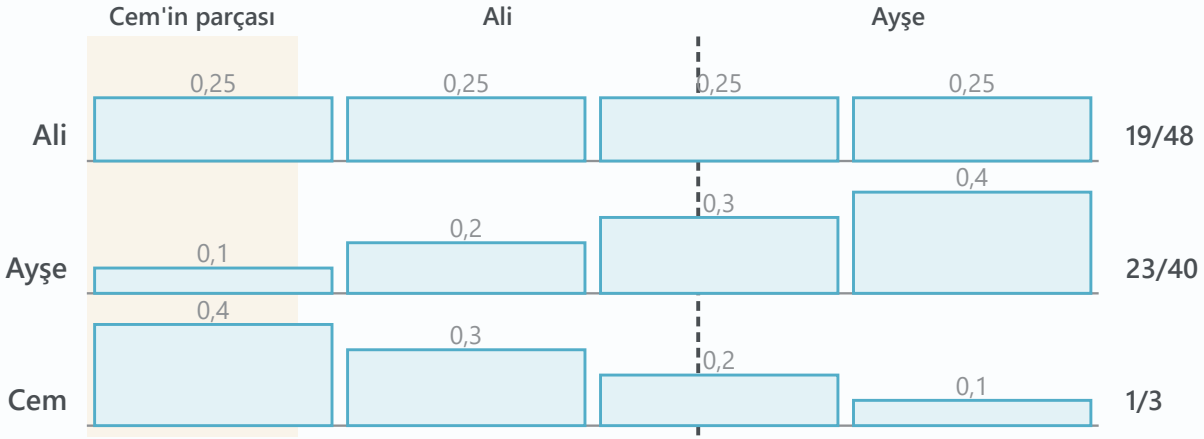
Cem için parça 1/2 > 1/3: fazlasını kırpar, kesim 5/24'e çekilir (Cem'in gözünde tam 1/3).

Son Azalan (Last Diminisher) Adım Adım



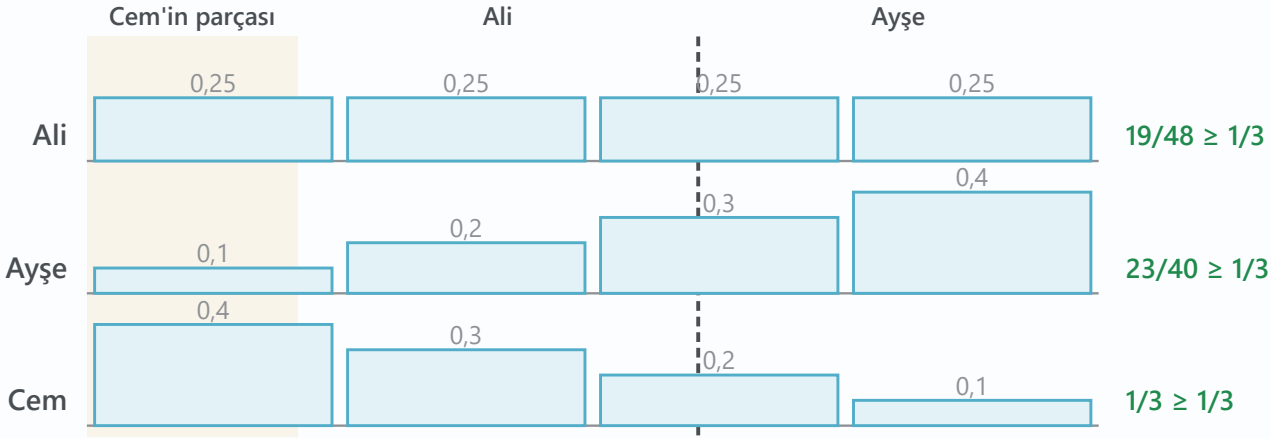
Parçayı **son kırpan** (Cem) alır ve oyundan çıkar. Kırpmayanlar için parça ≤ 1/3 olduğundan kalan pasta onların gözünde ≥ 2/3.

Son Azalan (Last Diminisher) Adım Adım



Kalan [5/24, 1] iki kişide kes-ve-seç: Ali 29/48'den ortadan keser, Ayşe değeri büyük olan sağı (23/40) seçer.

Son Azalan (Last Diminisher) Adım Adım



Herkes kendi gözünde ≥ 1/3 aldı ⇒ **orantılı**. n kişide $O(n^2)$ sorgu. Ama kıskançsız değil: Cem, Ali'nin parçasına 9/20 değer verir (kendi 1/3).

Hareketli Bıçak (Dubins–Spanier) ve Even–Paz

Kural

Bir bıçak pastanın solundan sağa yavaşça kayar. Bıçağın solundaki parça birinin gözünde 1/n değerine ulaşınca o kişi "dur!" der, parçayı alır ve çıkar; bıçak kalanlarla sürer, sonuncu kişi kalanı alır (Dubins ve Spanier, 1961).

kişi	1. tur: v([0, x]) = 1/3	2. tur: v([5/24, x]) = 1/3	aldığı
Cem	**x = 5/24 ≈ 0,21 (dur!)**	—	[0, 5/24]: 1/3
Ali	x = 1/3 ≈ 0,33	**x = 13/24 ≈ 0,54 (dur!)**	[5/24, 13/24]: 1/3
Ayşe	x = 19/36 ≈ 0,53	x = 43/72 ≈ 0,60	[13/24, 1]: 13/20

Değerler son azalan örneğindeki gibi (çeyrek başına Ali 0,25; Ayşe 0,1–0,4; Cem 0,4–0,1). İlk parça son azalandakiyle aynı: kesimi en önce Cem durdurur. Herkes ≥ 1/3 alır ⇒ **orantılı**. Kıskançsız değil: Cem, Ali'nin parçasına 2/5 > 1/3 değer verir.

Even–Paz (1984): pastayı her adımda ikiye bölerek orantılı paylaşımı çok daha az soruyla bulur.

Bölünemez Mallar: Sırayla Seçim ve EF1

kişi	m1	m2	m3	m4	kendi	ötekinin
Ali	7	5	3	1	0	0
Ayşe	8	6	1	1	0	0

Bölünemez 4 mal, 2 kişi; tablo değerleri gösterir. Sırayla seçim (round-robin): her turda sıradaki kişi kalan en değerli malı alır.

Adım 1 / 5

Bölünemez mallarda kıskançsızlık her zaman mümkün değildir (tek araba, iki kardeş). Sırayla seçim her zaman EF1 bir paylaşım üretir.

Bölünemez Mallar: Sırayla Seçim ve EF1

kişi	m1	m2	m3	m4	kendi	ötekinin
Ali	**7**	5	3	1	7	0
Ayşe	8	6	1	1	0	8

Ali m1'i (7) alır.

Adım 2 / 5

Bölünemez mallarda kıskançsızlık her zaman mümkün değildir (tek araba, iki kardeş). Sırayla seçim her zaman EF1 bir paylaşım üretir.

Bölünemez Mallar: Sırayla Seçim ve EF1

kişi	m1	m2	m3	m4	kendi	ötekinin
Ali	**7**	5	3	1	7	5
Ayşe	8	**6**	1	1	6	8

Ayşe kalanlardan m2'yi (6) alır.

Bölünemez mallarda kıskançsızlık her zaman mümkün değildir (tek araba, iki kardeş). Sırayla seçim her zaman EF1 bir paylaşım üretir.

Bölünemez Mallar: Sırayla Seçim ve EF1

kişi	m1	m2	m3	m4	kendi	ötekinin
Ali	**7**	5	**3**	1	10	5
Ayşe	8	**6**	1	1	6	9

Ali m3'ü (3) alır.

Bölünemez mallarda kıskançsızlık her zaman mümkün değildir (tek araba, iki kardeş). Sırayla seçim her zaman EF1 bir paylaşım üretir.

Bölünemez Mallar: Sırayla Seçim ve EF1

kişi	m1	m2	m3	m4	kendi	ötekinin
Ali	**7**	5	**3**	1	10	6
Ayşe	8	**6**	1	**1**	7	9

Ayşe m4'ü alır. Ayşe: kendi 7, Ali'nin demeti gözünde 9 ⇒ kıskanır! Ama Ali'nin demetinden m1 çıkarılırsa 1 ≤ 7: **EF1** (bir mala kadar kıskançsız).

Bölünemez mallarda kıskançsızlık her zaman mümkün değildir (tek araba, iki kardeş). Sırayla seçim her zaman EF1 bir paylaşım üretir.

Oyuna Dönüş: Adil Bölüşüm Oyunu

Açgözlü seçim en iyi tepki değil

Açılıştaki mallar (O1 / O2 değeri): Pasta 5/1, Kitap 4/9, Kulaklık 3/8, Top 1/1.

O1'in planı	Seçim sırası	O1	O2
hep en değerlisi	Pasta, Kitap, Kulaklık, Top	8	10
önce Kitap	Kitap, Kulaklık, Pasta, Top	**9**	9

O2 Pastaya 1 değer verir, Pasta beklese de kaçmaz. Sırayla seçimde her turda en sevdiğin malı almak her zaman en iyi strateji değildir.

Adalet ne oldu?

İkinci planda O2, O1'in demetine 9 + 1 = 10 değer verir, kendi demetine 9: **kıskanır**. Kitap çıkarılırsa 1 ≤ 9: paylaşım yine EF1'dir. Kuramın güvencesi: iki taraf da hep favorisini alırsa sırayla seçim EF1 üretir.

Gerçekte: Harvard İşletme Okulu'nun sırayla seçime (draft) dayalı ders dağıtımında öğrencilerin tam bu türden stratejik seçimler yaptığı ve bunun refah kaybına yol açtığı veriyle gösterildi (Budish ve Cantillon, 2012).

Adil bir protokol bile stratejik oyunculara açıktır: EF1 güvencesi, herkes sırası gelince favorisini alırsa geçerlidir.

Hangi Paylaşım? Üç Sosyal Refah Ölçütü

Faydacı
Sosyal refah SW = Σ v_i en büyük: verimlilik.

Eşitlikçi
En küçük v_i en büyük (maksimin).

Nash sosyal refahı
$v_{Ali} \times v_{Ayşe}$ en büyük; birinin değerleri 10 katına çıksa da sonuç aynı.

Paylaşım	Ali	Ayşe	toplam	en küçük	çarpım	kıskançsız?
Sırayla seçim: Ali {m1, m3}, Ayşe {m2, m4}	10	7	17	7	70	hayır, EF1
Faydacı: Ali {m3, m4}, Ayşe {m1, m2}	4	14	**18**	4	56	hayır, EF1 bile değil
Nash / eşitlikçi (tek değil): Ali {m2, m3, m4}, Ayşe {m1}	9	8	17	**8**	**72**	evet

Faydacıda Ali, Ayşe'nin demetine 12 verir (kendi 4); m1 çıkarılsa bile 5 > 4. Teorem (Caragiannis vd., 2016): toplamalı değerlerde Nash sosyal refahını maksimize eden paylaşım hem **Pareto optimal** hem **EF1**'dir.

Adaletin Bedeli ve Pareto Verimliliği

Pareto verimliliği

Başka bir paylaşım kimseyi kötüleştirmeden birini iyileştiremiyorsa paylaşım Pareto verimlidir (Pareto optimal). Örnekteki 16 paylaşımın 11'i Pareto optimal; sırayla seçimin (10, 7) paylaşımı da.

Adaletin bedeli (price of fairness)

En büyük sosyal refahın, adil paylaşımlar arasındaki en büyük sosyal refaha oranı. Oran 1'e yakınsa adalet ucuzdur.

Paylaşım	Ali	Ayşe	toplam	kıskançsız?
Ali {m3, m4}, Ayşe {m1, m2}	4	14	**18**	hayır
Ali {m3}, Ayşe {m1, m2, m4}	3	15	**18**	hayır
Ali {m2, m3, m4}, Ayşe {m1}	9	8	17	**evet**
Ali {m2, m3}, Ayşe {m1, m4}	8	9	17	**evet**

16 paylaşımdan yalnız ikisi kıskançsız; m4'teki eşitlik yüzünden ikisi de Nash ve eşitlikçi optimumdur, toplamları 17. Adaletin bedeli = 18 / 17 ≈ **1,059**: kıskançsızlık için sosyal refahın yaklaşık %5,6'sı feda edilir. Anarşinin Bedeli gibi bir orandır; kayıp bencillikten değil adalet kısıtından gelir. Bölünebilir pastada n kişide orantılılığın bedeli kabaca √n ile orantılı büyür (Caragiannis vd., 2012).

ETİK VE YÖNETİŞİM

Etik İlkeler ve Standartlar

Şeffaflık ve açıklanabilirlik
Karar süreçleri anlaşılır olmalı; kişi kararın gerekçesini öğrenebilmeli.

Hesap verebilirlik
Hatalı bir kararın sorumlusu belli olmalı; itiraz ve insan incelemesi mümkün olmalı.

Adalet ve eşitlik
Sosyal etkiler ve grup bazlı sonuçlar göz önünde tutulmalı.

Mahremiyet
Veri en az gerekli ölçüde ve amacına uygun toplanmalı.

GDPR (AB, 2018), md. 22	Yalnızca otomatik işlemeye dayanan önemli kararlara itiraz ve insan müdahalesi hakkı
AB Yapay Zekâ Yasası (2024)	Risk tabanlı: kredi skorlama ve işe alım "yüksek risk" sınıfında
KVKK (Türkiye), OECD YZ ilkeleri	Kişisel veri işleme kuralları; güvenilir yapay zekâ için ortak ilkeler

ETİK VE YÖNETİŞİM

Stratejik Tasarım ve Teşvikler

- Algoritma bir kural koyduğunda insanlar ona **tepki verir**: sınıflandırılan kişiler de oyuncudur (stratejik sınıflandırma).
- **Goodhart yasası:** bir ölçü hedef hâline gelince iyi bir ölçü olmaktan çıkar (tıklama sayısı → tıklama tuzağı).
- Kullanıcılara doğru teşvikler vermek ve kötü niyetli manipülasyonu (sahte yorum, bot hesap) önlemek tasarımın parçasıdır.

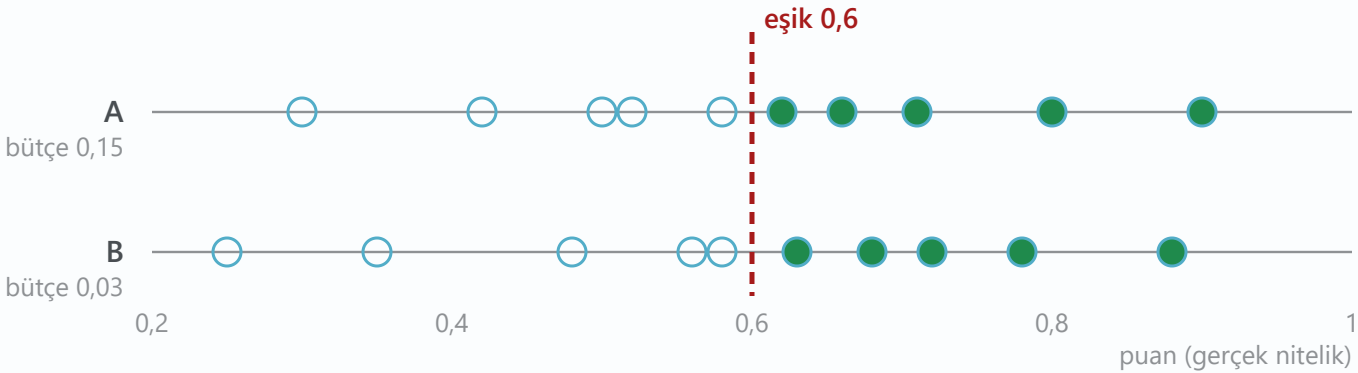
Oyunlanabilir özellik
Kredi puanında "kart sayısı" gibi kolay değiştirilen özellikler, gerçek riski değiştirmeden puanı değiştirir.

Öneri sistemleri
Etkileşimi değil kullanıcının uzun vadeli yararını ödüllendiren hedef.

Maliyet eşitsizliği
Oyunlama maliyeti gruplar arasında farklıysa stratejik davranış yeni bir adaletsizlik doğurur.

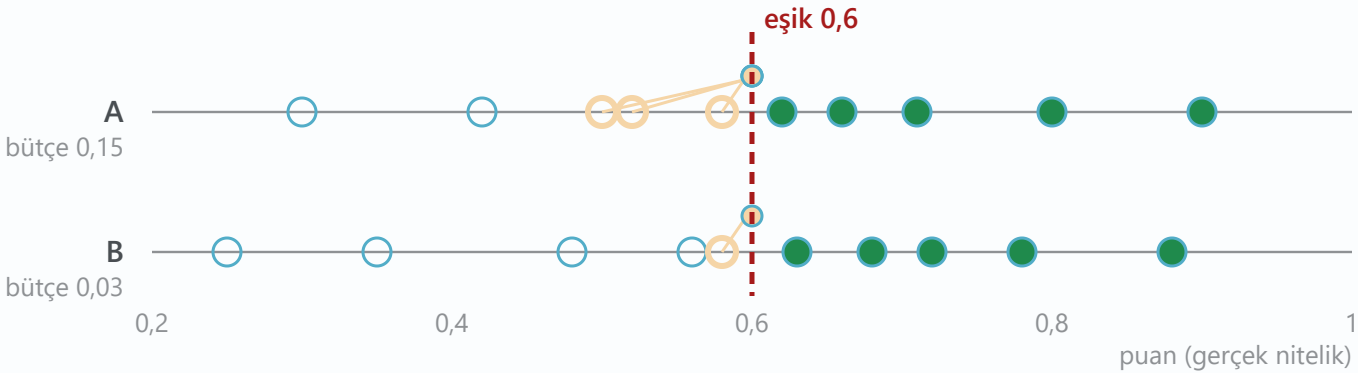
Mekanizma tasarımı sorusu burada da geçerli: kurala uyan dürüst davranış, bireyin çıkarına olmalıdır.

Stratejik Sınıflandırma Adım Adım



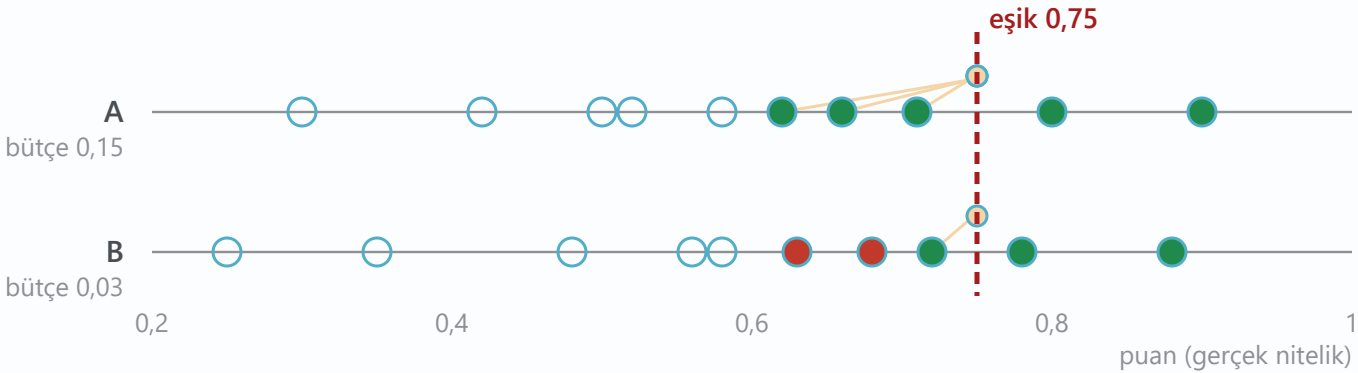
Eşik 0,6; dolu daire = gerçekten nitelikli (puan ≥ 0,6). Kişiler puanlarını bir **maliyetle** yükseltebilir (ör. kart sayısı). A'nın bütçesi 0,15, B'nin 0,03.

Stratejik Sınıflandırma Adım Adım



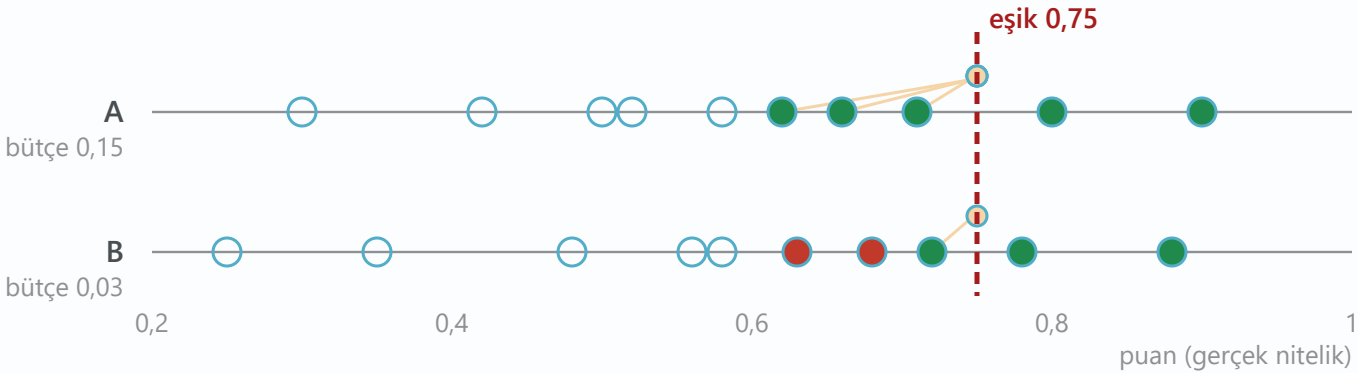
Oyunlama: eşiğe en fazla bütçe kadar yakın olanlar puanını eşiğe taşır. A'dan 3, B'den 1 niteliksiz kişi geçer (turuncu halka).

Stratejik Sınıflandırma Adım Adım



Kurum tepki verir: eşiği 0,75'e yükseltir (A'nın bütçesini hesaba katarak). Artık hiçbir niteliksiz kişi geçemez.

Stratejik Sınıflandırma Adım Adım



Ama bütçesi küçük B'nin nitelikli 5 kişisinden yalnız 3'ü geçer (kırmızı): A'da 5/5. Stratejik davranışa karşı alınan önlem yeni bir **eşitsizlik** doğurdu (Hu, Immorlica, Vaughan 2019).

ETİK VE YÖNETİŞİM

Algoritmik Denetim

Ne denetlenir?
Veri kaynağı, hedef değişken, grup bazlı performans ve insan gözetimi birlikte incelenir.

Belgeleme
Model kartları ve veri kümesi kartları varsayımları, kullanım amacını ve bilinen sınırlamaları görünür kılar.

Dağıtım sonrası
Veri kayması, itirazlar ve beklenmeyen zararlar düzenli olarak izlenir.

Katılım
Etkilenen grupların görüşü alınır; itiraz hakkı ve açık bir sorumluluk zinciri tanımlanır.

Denetim listesi: taban oranları, seçim oranı, TPR, FPR, PPV ve kalibrasyon **her grup için ayrı ayrı** raporlanır; hangi ölçütün neden önceliklendirildiği yazılı gerekçelendirilir.

Tartışma: Algoritmalar Tarafsız Olabilir mi?

- Algoritmalar hedef, veri, ölçüt ve eşik seçimlerini yapan kurumların değerlerinden bağımsız değildir.
- Tarihsel eşitsizlikler eğitim verisine ve hedef değişkenlere yansıyabilir.
- Farklı adalet ölçütleri aynı karar için çelişen sonuçlar üretir (uyumsuzluk teoremleri).
- Model, dağıtımdan sonra davranışı değiştirerek yeni geri besleme döngüleri yaratabilir.

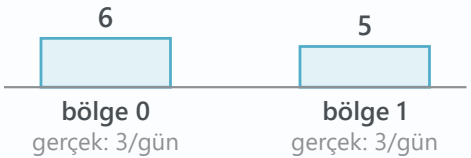
Temel zorluklar

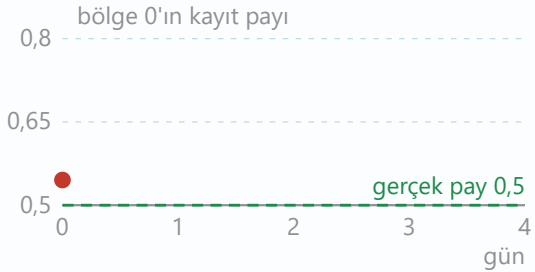
Adalet kavramının bağlama göre değişmesi; doğruluk, mahremiyet ve grup adaleti arasındaki ödünleşimler; kültürel farklılıklar.

Yanıt

Mutlak tarafsızlık iddiası yerine etkileri ölçmek, tercihleri belgelemek, itiraz mekanizması kurmak.

Kod İzleme: Kendini Doğrulayan Veri





İki bölgede gerçek olay sayısı **eşit**; geçmiş kayıtlarda bölge 0 bir fazla (6 ile 5), belki tesadüf.

Adım 1 / 9

```
▶ kayit = [6, 5]      # geçmiş kayıtlar (bölge 0, 1)
▶ gercek = [3, 3]     # gerçek günlük olay: eşit!
3 for gun in range(1, 5):
4   hedef = 0 if kayit[0] >= kayit[1] else 1
5   kayit[hedef] += gercek[hedef]  # yalnız orada
6   pay = kayit[0] / sum(kayit)
7   print(gun, hedef, kayit, round(pay, 2))
```

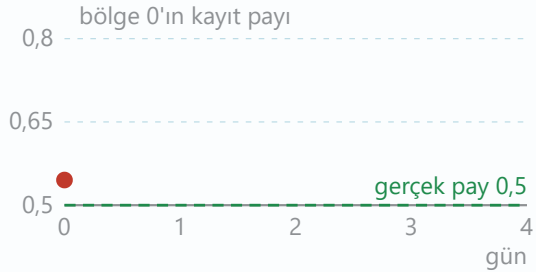
DEĞİŞKENLER

`kayit = [6, 5]` `gercek = [3, 3]`

ÇIKTI

Kod İzleme: Kendini Doğrulayan Veri





Gün 1: devriye kaydı fazla olan bölgeye gider ⇒ hedef = 0.

Adım 2 / 9

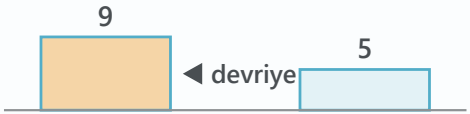
```
1 kayit = [6, 5]     # geçmiş kayıtlar (bölge 0, 1)
2 gercek = [3, 3]    # gerçek günlük olay: eşit!
▶ for gun in range(1, 5):
▶   hedef = 0 if kayit[0] >= kayit[1] else 1
5   kayit[hedef] += gercek[hedef]  # yalnız orada
6   pay = kayit[0] / sum(kayit)
7   print(gun, hedef, kayit, round(pay, 2))
```

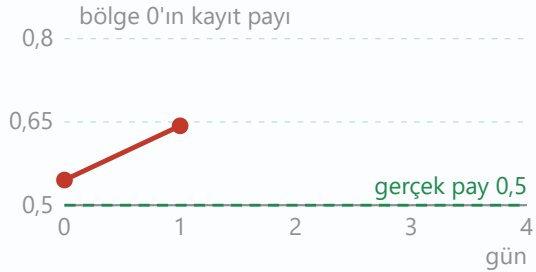
DEĞİŞKENLER

`gun = 1` `kayit = [6, 5]` `hedef = 0`

ÇIKTI

Kod İzleme: Kendini Doğrulayan Veri





Yalnız devriyenin gittiği yerde olay **kaydedilir**: kayit = [9, 5]. Bölge 1'deki 3 olay görünmez kalır. Pay 0,64.

Adım 3 / 9

```
1 kayit = [6, 5]     # geçmiş kayıtlar (bölge 0, 1)
2 gercek = [3, 3]    # gerçek günlük olay: eşit!
3 for gun in range(1, 5):
4   hedef = 0 if kayit[0] >= kayit[1] else 1
▶   kayit[hedef] += gercek[hedef]  # yalnız orada
▶   pay = kayit[0] / sum(kayit)
▶   print(gun, hedef, kayit, round(pay, 2))
```

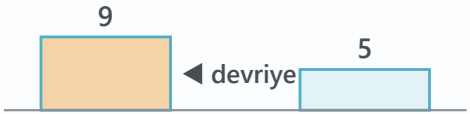
DEĞİŞKENLER

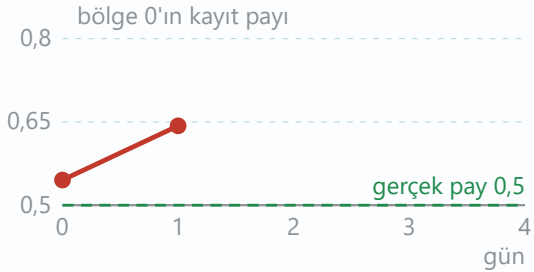
`kayit = [9, 5]` `pay = 0.643`

ÇIKTI

```
1 0 [9, 5] 0.64
```

Kod İzleme: Kendini Doğrulayan Veri





Gün 2: devriye kaydı fazla olan bölgeye gider ⇒ hedef = 0.

Adım 4 / 9

```
1 kayit = [6, 5]     # geçmiş kayıtlar (bölge 0, 1)
2 gercek = [3, 3]    # gerçek günlük olay: eşit!
▶ for gun in range(1, 5):
▶   hedef = 0 if kayit[0] >= kayit[1] else 1
5   kayit[hedef] += gercek[hedef]  # yalnız orada
6   pay = kayit[0] / sum(kayit)
7   print(gun, hedef, kayit, round(pay, 2))
```

DEĞİŞKENLER

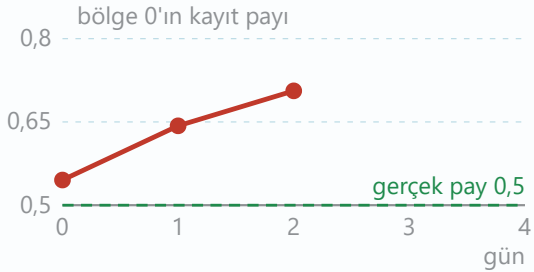
`gun = 2` `kayit = [9, 5]` `hedef = 0`

ÇIKTI

```
1 0 [9, 5] 0.64
```

Kod İzleme: Kendini Doğrulayan Veri





Yalnız devriyenin gittiği yerde olay **kaydedilir**: kayit = [12, 5]. Bölge 1'deki 3 olay görünmez kalır. Pay 0,71.

Adım 5 / 9

```
1 kayit = [6, 5]     # geçmiş kayıtlar (bölge 0, 1)
2 gercek = [3, 3]    # gerçek günlük olay: eşit!
3 for gun in range(1, 5):
4   hedef = 0 if kayit[0] >= kayit[1] else 1
▶   kayit[hedef] += gercek[hedef]  # yalnız orada
▶   pay = kayit[0] / sum(kayit)
▶   print(gun, hedef, kayit, round(pay, 2))
```

DEĞİŞKENLER

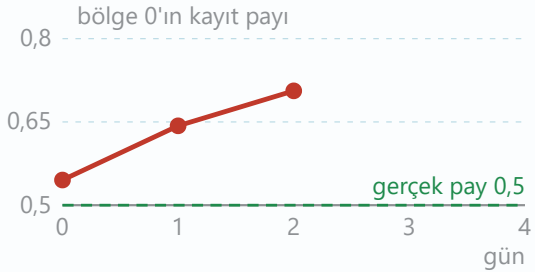
`kayit = [12, 5]` `pay = 0.706`

ÇIKTI

```
1 0 [9, 5] 0.64
2 0 [12, 5] 0.71
```

Kod İzleme: Kendini Doğrulayan Veri





Gün 3: devriye kaydı fazla olan bölgeye gider ⇒ hedef = 0.

Adım 6 / 9

```
1 kayit = [6, 5]     # geçmiş kayıtlar (bölge 0, 1)
2 gercek = [3, 3]    # gerçek günlük olay: eşit!
▶ for gun in range(1, 5):
▶   hedef = 0 if kayit[0] >= kayit[1] else 1
5   kayit[hedef] += gercek[hedef]  # yalnız orada
6   pay = kayit[0] / sum(kayit)
7   print(gun, hedef, kayit, round(pay, 2))
```

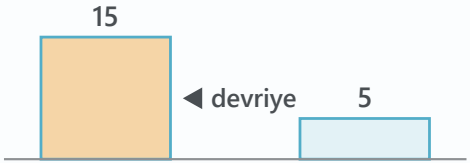
DEĞİŞKENLER

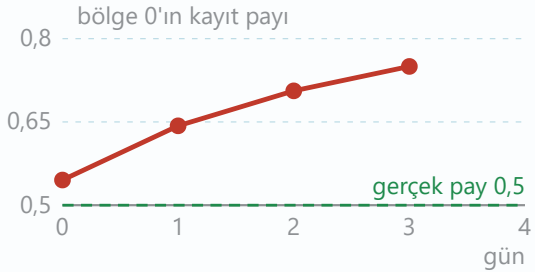
`gun = 3` `kayit = [12, 5]` `hedef = 0`

ÇIKTI

```
1 0 [9, 5] 0.64
2 0 [12, 5] 0.71
```

Kod İzleme: Kendini Doğrulayan Veri





Yalnız devriyenin gittiği yerde olay **kaydedilir**: kayit = [15, 5]. Bölge 1'deki 3 olay görünmez kalır. Pay 0,75.

Adım 7 / 9

```
1 kayit = [6, 5]     # geçmiş kayıtlar (bölge 0, 1)
2 gercek = [3, 3]    # gerçek günlük olay: eşit!
3 for gun in range(1, 5):
4   hedef = 0 if kayit[0] >= kayit[1] else 1
▶   kayit[hedef] += gercek[hedef]  # yalnız orada
▶   pay = kayit[0] / sum(kayit)
▶   print(gun, hedef, kayit, round(pay, 2))
```

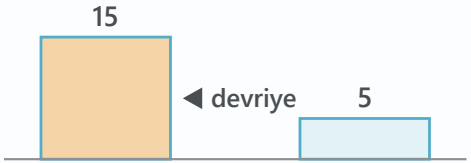
DEĞİŞKENLER

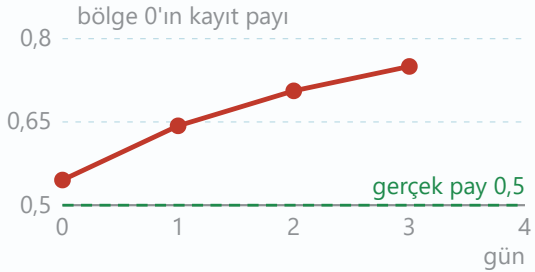
`kayit = [15, 5]` `pay = 0.75`

ÇIKTI

```
1 0 [9, 5] 0.64
2 0 [12, 5] 0.71
3 0 [15, 5] 0.75
```

Kod İzleme: Kendini Doğrulayan Veri





Gün 4: devriye kaydı fazla olan bölgeye gider ⇒ hedef = 0.

Adım 8 / 9

```
1 kayit = [6, 5]     # geçmiş kayıtlar (bölge 0, 1)
2 gercek = [3, 3]    # gerçek günlük olay: eşit!
▶ for gun in range(1, 5):
▶   hedef = 0 if kayit[0] >= kayit[1] else 1
5   kayit[hedef] += gercek[hedef]  # yalnız orada
6   pay = kayit[0] / sum(kayit)
7   print(gun, hedef, kayit, round(pay, 2))
```

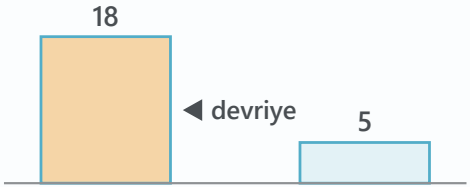
DEĞİŞKENLER

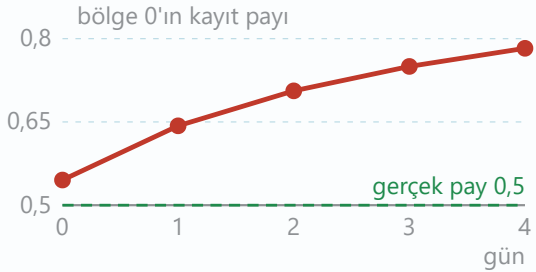
`gun = 4` `kayit = [15, 5]` `hedef = 0`

ÇIKTI

```
1 0 [9, 5] 0.64
2 0 [12, 5] 0.71
3 0 [15, 5] 0.75
```

Kod İzleme: Kendini Doğrulayan Veri





Pay 0,55 → 0,78: veri, modelin kararıyla toplandığı için modeli "doğrular" (Ensign vd., 2018).

Adım 9 / 9

```
1 kayit = [6, 5]     # geçmiş kayıtlar (bölge 0, 1)
2 gercek = [3, 3]    # gerçek günlük olay: eşit!
3 for gun in range(1, 5):
4   hedef = 0 if kayit[0] >= kayit[1] else 1
▶   kayit[hedef] += gercek[hedef]  # yalnız orada
▶   pay = kayit[0] / sum(kayit)
▶   print(gun, hedef, kayit, round(pay, 2))
```

DEĞİŞKENLER

`kayit = [18, 5]` `pay = 0.783`

ÇIKTI

```
1 0 [9, 5] 0.64
2 0 [12, 5] 0.71
3 0 [15, 5] 0.75
4 0 [18, 5] 0.78
```

Gerçek Hayat Uygulamaları ve Etkinlik

Kredi ve bankacılık
Skorlama; kalibrasyon ve fırsat eşitliği arasındaki seçim.

İşe alım ve İK
Özgeçmiş eleme; tarihsel veri önyargısı, beşte dört kuralı.

Sağlık
Teşhis ve tedavi önerisi; vekil hedef (harcama ≠ ihtiyaç).

Etkinlik: Küçük bir veri kümesi için eşik tabanlı bir sınıflandırıcı tasarlayın. (1) İki grubun karışıklık matrisini çıkarın. (2) Dört grup oranını (seçim oranı, TPR, FPR, PPV) hesaplayın (`olcutler` fonksiyonu). (3) Eşiği değiştirerek hangi ölçütü sağlayabildiğinizi ve neyi kaybettiğinizi raporlayın.

ÖZET

Özet

Ölçütler

Demografik eşitlik (seçim oranı), eşitlenmiş olasılıklar (TPR, FPR), tahmin eşitliği (PPV), kalibrasyon.

Örneğimiz

Kalibre ve PPV eşit bir model; buna rağmen seçim oranı 0,5/0,2 ve TPR 0,8/0,5.

Uyumsuzluk

FPR = $p/(1-p)\cdot(1-PPV)/PPV\cdot TPR$: taban oranı farklıysa ölçütler birlikte sağlanamaz (Chouldechova; Kleinberg vd.).

Adil bölüşüm

Kes-ve-seç: 2 kişide orantılı ve kıskançsız. 3+ kişide orantılı ≠ kıskançsız; bölünemezde EF1.

Etik

Şeffaflık, hesap verebilirlik, mahremiyet; GDPR, AB YZ Yasası.

Teşvikler

Algoritmaya tepki veren insanlar: Goodhart yasası, stratejik sınıflandırma.

ÖZET

Etkileşimli Oyunlar

Adil Bölüşüm Oyunu

Bölünemez malların sırayla seçimle paylaşımı: açgözlü ve stratejik seçim, kıskançsızlık ve EF1 denetimi.

ÖZET

Örnek Problemler

1. Grup C: TP = 30, FP = 20, FN = 10, TN = 40. Seçim oranı, TPR, FPR ve PPV?
2. 1. sorudaki C grubu ile bu bölümdeki A grubu arasında hangi ölçütler sağlanır?
3. p = 0,2, PPV = 0,5, TPR = 0,6 ise FPR kaçtır?
4. Kes-ve-seçte Ali kesiyor ve Ayşe'nin çikolatayı 3 kat sevdiğini biliyor. Ayşe iki parçayı eşit değerli bulursa çikolatalı (sağ) parçayı seçer. Ali nereden keser, kendi gözünde ne kazanır?
5. İki kişilik paylaşımda orantılılığın kıskançsızlığı gerektirdiğini gösterin.
6. Sırayla seçimde Ayşe başlasaydı (aynı değerlerle) kim neyi alırdı? Bir kişi iki malı eşit değerli bulursa numarası küçük olanı alır. Paylaşım kıskançsız mı?

▶ **Cevapları göster**

ÖZET

Alıştırmalar: Yeni Konular

1. `b_grubu` izlemesinde A'nın TPR'si 0,7 olsaydı B'de fırsat eşitliği için q kaç olmalıydı? B'nin doğruluğu ne olurdu?
2. `kes_noktasi(v_ayse, 0.5, 3)` çalışınca döngü sonunda [lo, hi] nedir? Döndürülen orta noktanın gerçek kesim noktasından uzaklığının kesinlikle 0,001'in altında olması için en az kaç sorgu gerekir?
3. Son azalan örneğinde Cem'in değerleri Ayşe'ninki gibi (0,1; 0,2; 0,3; 0,4) olsaydı ilk parçayı kim alırdı?
4. L = 2 ve d(x, y) = 0,1 iken f(x) = 0,8 ise bireysel adalet f(y) için hangi aralığı izin verir?
5. Geri besleme kodunu `kayit = [5, 6]` ile çalıştırın. 4 gün sonra bölge 0'ın payı kaçtır?

▶ **Cevapları göster**

Kaynaklar (1/2)

- S. Barocas, M. Hardt, A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities*, MIT Press, 2023.
- A. Chouldechova, "Fair prediction with disparate impact: a study of bias in recidivism prediction instruments", *Big Data* 5(2), 2017.
- J. Kleinberg, S. Mullainathan, M. Raghavan, "Inherent trade-offs in the fair determination of risk scores", *ITCS*, 2017.
- M. Hardt, E. Price, N. Srebro, "Equality of opportunity in supervised learning", *NeurIPS*, 2016.
- J. Angwin, J. Larson, S. Mattu, L. Kirchner, "Machine Bias", *ProPublica*, 2016.
- J. Buolamwini, T. Gebru, "Gender Shades", *Proc. FAT**, 2018.
- S. J. Brams, A. D. Taylor, *Fair Division: From Cake-Cutting to Dispute Resolution*, Cambridge University Press, 1996.
- C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, "Fairness through awareness", *ITCS*, 2012.

Kaynaklar (2/2)

- D. Ensign vd., "Runaway feedback loops in predictive policing", *Proc. FAT**, 2018.
- L. Hu, N. Immorlica, J. W. Vaughan, "The disparate effects of strategic manipulation", *Proc. FAT**, 2019.
- A. D. Procaccia, "Cake cutting algorithms", *Handbook of Computational Social Choice* içinde, Bölüm 13, 2016.
- I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, J. Wang, "The unreasonable fairness of maximum Nash welfare", *ACM EC*, 2016.
- A. R. Karlin, Y. Peres, *Game Theory, Alive*, AMS, 2017 (adil bölüşüm bölümü). Ders sayfaları: Procaccia, CMU 15-896 · Shah, Toronto CSC304.
- E. Budish, E. Cantillon, "The multi-unit assignment problem: theory and evidence from course allocation at Harvard", *American Economic Review* 102(5), 2012.
- H. Aziz, S. Mackenzie, "A discrete and bounded envy-free cake cutting protocol for any number of agents", *FOCS*, 2016.
- L. E. Dubins, E. H. Spanier, "How to cut a cake fairly", *American Mathematical Monthly* 68(1), 1961; S. Even, A. Paz, "A note on cake cutting", *Discrete Applied Mathematics* 7(3), 1984.
- I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, M. Kyropoulou, "The efficiency of fair division", *Theory of Computing Systems* 50(4), 2012.

ÖZET

Kaynaklar: Mevzuat ve Haberler

- J. Dastin, "Amazon scraps secret AI recruiting tool that showed bias against women", *Reuters*, 10 Ekim 2018.
- Avrupa Parlamentosu ve Konseyi, Genel Veri Koruma Tüzüğü (GDPR), Tüzük (AB) 2016/679, md. 22 (uygulama 2018).
- Avrupa Parlamentosu ve Konseyi, Yapay Zekâ Yasası, Tüzük (AB) 2024/1689.
- 6698 sayılı Kişisel Verilerin Korunması Kanunu (KVKK), Resmî Gazete, 7 Nisan 2016.
- OECD, *Recommendation of the Council on Artificial Intelligence* (OECD YZ ilkeleri), 2019.

Son

Bölüm 11: Algoritmik Adalet ve Etik

Sonraki bölüm: Bölüm 12: Davranışsal ve Kolektif Karar

Çalışma soruları · Terim sözlüğü