KOÜ BİLGİSAYAR MÜHENDİSLİĞİ

BÜYÜK VERİ ANALİZİNE GİRİŞ FİNAL			
Kod : *BLM 442* Akad. yıl : *2018-2019* Sömestr : *BAHAR* Tarih : *18.06.2019* Baş. Saati: *15:00* Süre : *90 dk*	Soyad : İsim : Öğrenci no: İmza :	Sıra no:	
	Toplam 6 sayfada 8 Soru 120 puan üzerinden		

1. ( )	2. ( )	3. ( )	4. ( )	5. ( )	6. ( )	7. ( )	8. ( )		
12	20	13	10	20	15	20	10		120

Not: Kod parçaları için gerekli kütüphanelerin import edildiğini düşünerek cevaplayınız. Kod çıktı sorularında tam olmayan cevaplar puanlandırılmayacaktır.

1. (Lineer cebir, istatistik, olasık temeller)

1.1. *(6p)* M matrisi ve v vektörleri verilmiş olsun. Aşağıdaki kod parçasının verilen değerler için çıktısı ne olur?

$$M = \begin{bmatrix} 3 & 0 & 2 \\ 2 & 0 & -2 \\ 0 & 1 & 1 \end{bmatrix} \quad v^T = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

```
print M.dot(v)
print v.dot(v)
print v.T.dot(v)
```

→ [[9]
[-4]
[5]]

→ [[4]]

1) Value Error verir. (3,1) ve (3,1) boyutlarının uyuşmamasından dolayı.

1.2. *(6p)* Rassal değişken X üzerinde BVeri isminde bir fonksiyon aşağıdaki gibi tanımlanmıştır. $BVeri_X(x) = P(X <= x)$. P ise rassal X değişkeninin alabileceği değerlerden birinin yani küçük x'e kadar olan olasılıkların toplamını göstermektedir. Örneğin X rassalı, bir madenin paranın havaya üç kere arka arkaya atılmasındaki "Tura" adedini göstermiş olsun. Buna göre $BVeri_X(0)$ ve $BVeri_X(2)$ değerlerini hesaplayınız?

Sample Space = S = {TTT, TTY, TYT, YTT, TYY, YTY, YYT, YYY}

Rastgele değişken X tura adedini gösteriyorsa;

X = S → ℝ X(TTT) = 3
X(TTY) = X(TYT) = X(YTT) = 2
X(TYY) = X(YTY) = X(YYT) = 1
X(YYY) = 0

P(X=0) = 1/8
P(X=1) = 3/8
P(X=2) = 3/8
P(X=3) = 1/8

BVeri_X(0) = P(X<=0) = P(X=0) = 1/8

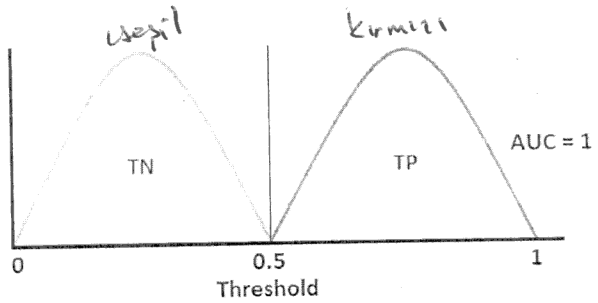
BVeri_X(2) = P(X<=2) = P(X=2) + P(X=1) + P(X=0) = 3/8 + 3/8 + 1/8 = 7/8

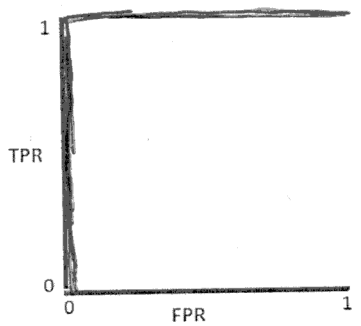
2. (Model değerlendirme (evaluation))

2.1. *(4p)* Bir modelin performansını neden değerlendiriyoruz?

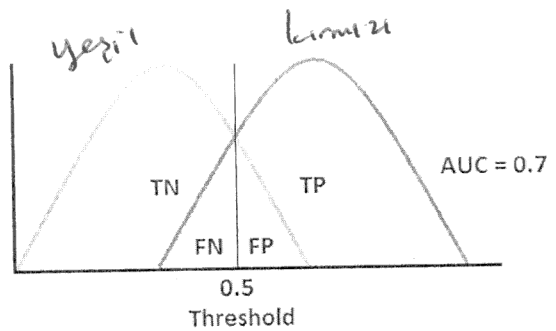
- En iyi performans gösteren modelleri bulmak,
- Parametre ayarının (parameter tuning) bir parçası,
- Sonuçları raporlama / yayınlama
- Araştırma/iş kararlarını nerede

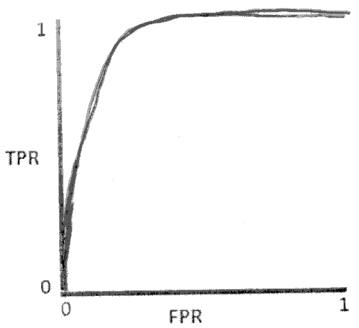
2.2. *(8p)* ROC eğrisi, farklı sınıflandırma eşikleri (threshold) için yanlış pozitif orana (FPR-false positive rate) karşılık gerçek pozitif oranın (TPR-true positive rate) bir grafiğidir. Soruda,kırmızı dağılım eğrisi pozitif sınıfa (hastalığı olan hastalar) ve yeşil dağılım eğrisi ise negatif sınıfa (hastalığı olmayan hastalar) karşılık gelmektedir. Farklı sınıflandırma eşikleri için yanlarına ROC grafiklerini çiziniz (ipucu: AUC-area under the curve).



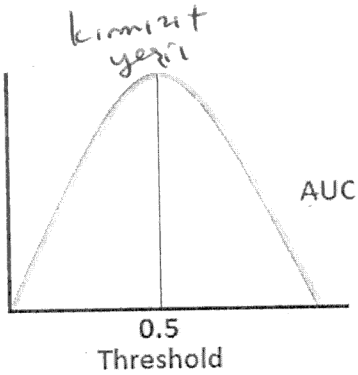


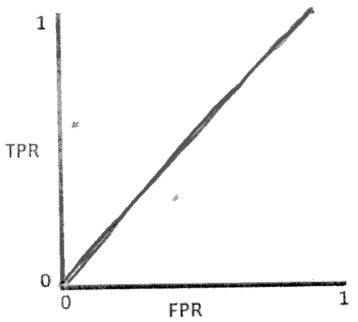
2p



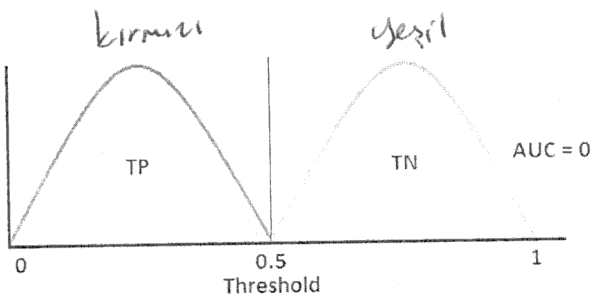


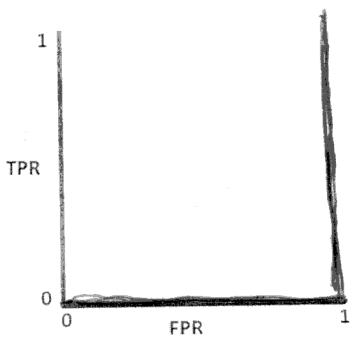
2p





2p





2p

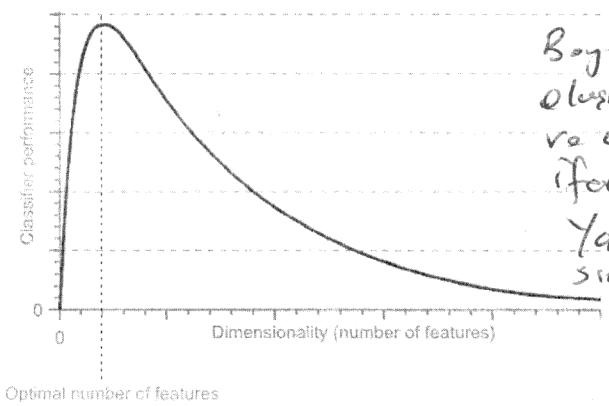
2.3. *(8p)* Özelliklerin 60x10'luk bir X matrisinde ve etiketlerin 60x1'lik by y vektöründe tutulduğu her biri 10 özelliğe (features) sahip 60 eğitim örneğinin (samples) bulunduğu denetimli bir regresyon problemi düşünün. Aşağıda gösterildiği gibi k parametresine sahip bir modeliniz olduğunu varsayın. k değeri 1, 2 veya 3 olabilir. 3 kat çapraz doğrulama (3-fold cross validation) kullanarak k'nın nasıl seçileceğini açıklayan sözde kodu verin.

- model = train(X, y, k); % k parametresi ile modeli {X, y} üzerinde eğit
- yhat = predict(model, Xhat); % modeli kullanarak Xhat üzerinde tahmin yap

- X (özellik vektörü) ve y (sınıf etiketini) şu şekilde 3'eşiyorum.
  X_1 ve y_1 → ilk 20 eğitim seti
  X_2 ve y_2 → 21-40 arasındaki eğitim seti
  X_3 ve y_3 → 41-60 arasındaki eğitim seti.) 2p
- k, 1'den 3'e kadar her bir fold hesabını yapıyorum.
  - Fold 1 =
    - Xtrain → {X_2, X_3} ytrain → {y_2, y_3}
    - model = train(Xtrain, ytrain, k)
    - yhat = predict(model, X_1)
    - err1 = yhat ve y_1 arasında karesel hata hesapla.) 2p
  - Fold 2 =
    - Xtrain → {X_1, X_3} ytrain → {y_1, y_3}
    - model = train(Xtrain, ytrain, k)
    - yhat = predict(model, X_2)
    - err2 = yhat ve y_2 arasında karesel hata hesapla) 2p
  - Fold 3 =
    - Xtrain → {X_1, X_2} ytrain → {y_1, y_2}
    - model = train(Xtrain, ytrain, k)
    - yhat = predict(model, X_3)
    - err3 = yhat ve y_3 arasında karesel hata hesapla) 2p
- err = (err1 + err2 + err3)/60
- En düşük hatayla sonuçlanan k'yı döndür.

3. (Boyutsallık ve Özellik Seçme)

3.1. *(4p)* Boyutsallık belası (curse of dimensionality) problemini aşağıdaki şekil üzerinden tartışınız?



Boyutluluk belası / laneti; düşük boyutlu ortamlarda oluşmayan yüksek boyutlu uzaylardaki nesneleri analiz ve organize ederken ortaya çıkan çeşitli olayları ifade eder.
Yandaki şekilde de feature sayısı arttıkça sınıflandırıcının performansının optimum sayıda feature'a ulaşana kadar arttığını gösterebiliriz.

3.2. *(4p)* Özellik çıkarma (extraction) ile özellik seçme (selection) arasındaki fark nedir bir cümle ile açıklayıp birer örnek yöntem veriniz? Temel olarak her iki yöntem yukarıdaki boyutsallık problemindeki boyutu azaltmak için kullanılmakla beraber özellik seçmede gerekli/tekrar eden özellikler filtrelenerek orijinal özelliklerden kopmadan azaltma yapılır. (forward/backward selection gibi) 2p
- özellik çıkarmada orijinal özelliklerden daha az boyutta farklı özellikler elde edilir (PCA gibi). 2p

3.3. *(5p)* Dengesiz (imbalanced) veri kümesi nedir, dengeli hale hangi yöntem(ler)le getirilir?

2.5p Veriseti dengesizliği genellikle bir veri kümesi içindeki eşit olmayan sınıf dağılımını gösterir. i- down-sampling ii) up-sampling 2.5p 2.5p - ensembling methods

4. (Scikit-learn)

4.1. *(3p)* Diyelim ki özellik olarak sehir_id'ye sahip bir veri kümeniz var, ne yapardınız?

Makine öğrenmesi projesi için veri toplanırken toplanan verilerden özellikler dikkatle belirlenmesi gerekir. Sehir_id sadece bir sıra numarasıdır. Aksi belirtilmedikçe şehirle ilgili bir özelliği temsil etmez. Dolayısıyla özellikler setinden drop etmek gerekir.

4.2. *(4p)* Normalleştirme ve standardizasyon arasındaki fark nedir?

2p Normalleştirme, bir özelliğin değerlerini bir aralığa ölçer/normalize eder. (min-max normalleştirme)

2p Standardizasyon; verinin ortalaması (mean) 0 (sıfır), standart sapması (σ) 1 (bir) olacak dağılıma transform eder.

4.3. *(3p)* Python'da herhangi bir makine öğrenmesi algoritmasını uygulamak için temel adımlar nelerdir?

Ham veri elde edilmesi → özellik çıkarma → öğrenme → değerlendirme → tahmin.

5. (Hadoop Tasarım Kalıpları)

5.1. *(15p)* Klasik "WordCount" MapReduce örneği aşağıdaki kodda verildiği gibidir.

```
class Mapper
    method Map(docid id, doc d)
        for all term t in doc d do
            Emit(term t, count 1)


class Reducer
    method Reduce(term t, counts [c1, c2,...])
        sum = 0
        for all count c in [c1, c2,...] do
            sum = sum + c
        Emit(term t, count sum)
```

Bütün dokumanları tarayıp içinde "Baskan X", "Baskan Y", "Baskan ..." geçen keyword'leri bulup daha sonra başkanları {X, Y, ...} şeklinde listeleyecek yukarıdaki koda benzer bir kod yazınız. Burada sayma işlemi yoktur, sadece benzersiz (unique) baskan isimleri bulunacaktır. (İpucu: Önce "Baskan" kelimesini bulun ve bir sonraki kelime son listeye gönderin.)

```
class Mapper
    method map(docid id, doc d)
        for all term t in doc d do
            if t.equals("Baskan")
                on_ad = true
            if on_ad
                Emit(term t, count 1)
                on_ad = false
```

10p

```
class Reducer
    method Reduce(term t,
                  counts [c1, c2, ...])
        Emit(term t, count 0)
```

5p

5.2. *(5p)* Tasarım kalıbı nedir ve kullanılmasının iki avantajını veriniz.

2.5p Yazılım tasarımında sıklıkla karşılaşılan sorunlara genel olarak tekrarlanabilir ve başarılı çözüm getiren hazır kalıplardır.

2.5p
- Test edilmiş, geliştirme katkısı kanıtlanmış yöntemler sürdüğü için geliştirme süreci hızlandırır.
- Kalıp [illegible] bakım maliyetini azaltır.

6. (Apache Spark)

6.1.*(5p)* RDD bağlamında dönüşümler ve aksiyonlar (transformations and actions) nedir, birer örnek metot veriniz.

2.5p Spark RDD dönüşümü var olan RDD'lerden yeni bir RDD üretir (map, filter gibi)
2.5p Aksiyonlar ise geriye değer dönüş olan operasyonlardır (count, collect gibi)

6.2.*(5p)* Diyelim ki HDFS'deki bir dosyanın her bir satırında bir sayı vardır. Bu sayıların karelerinin toplamının kare kökünü Spark kullanarak nasıl hesaplardın?

```
1p  #HDFS'deki sayilar.txt dosyasını RDD olarak yükle
    sayilarASText = sc.textFile("hdfs://ipcolab/user/sayilar.txt")

1p  #Kare hesaplamak için bir fonksiyon tanımla (kare(str))
    kare(str):
      v = int(str);
      return v*v;

1p  #Tanımladığın fonksiyonu Spark RDD üzerinde dönüşüm olarak çalıştır
    sayilar = sayilarASText.map(kare);

1p  #Kareler toplamını reduce aksiyonu ile bul
    toplam = sayilar.reduce(sum);

1p  #Toplamın kare kökünü hesapla (math.sqrt)
    math.sqrt(toplam);
```

6.3.*(5p)* Spark ekosistemi/kütüphaneleri nelerdir, ne amaçla kullanılır?

(i) Spark Streaming = Gerçek-zamanlı veri analizi
(ii) Spark SQL = SQL benzeri sorgularla spark jobları üzerinde çalışmayı sağlar
(iii) Spark MLLib = Makine öğrenmesi işlemlerinin yapılmasını sağlar
(iv) Spark GraphX = Çizge işlemleri yapmaktadır.

7. (NoSQL veritabanları ve Neo4j)

7.1.*(5p)* CAP teoremi nedir? NoSQL sistemlerine nasıl uygulanabilir?

2.5p CAP teoremi, dağıtık bir sistemin aynı anda tutarlılık (consistency), ulaşılabilirlik (availability) bölünebilme toleransı (partition tolerance) koşullarına aynı anda sahip olamayacağını söyler.
2.5p Bir NoSQL VT; AP sistem (Dynamo, CouchDB vs) veya CP sistem (MongoDB, HBase gibi) dağıtık bir ortamda sadece ikisini sağlanabilir. Bölünebilme toleransı dağıtık VT için elzemdir. Dolayısıyla

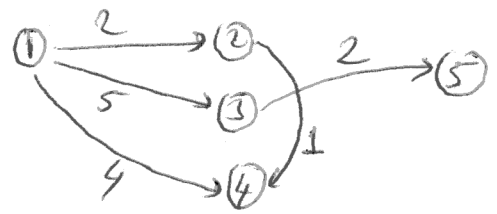
7.2.*(5p)* Neo4j'deki düğümler (Nodes), ilişkiler (Relationships), özellikler (Properties) ve etiketler (Labels) gibi yapı taşlarının rolünü açıklayınız?

Nodes – kayıtlar Relationship – Kayıtların ilişkileri
Properties – Node'ların özellikleri / içerdikleri veriler
Labels – Node gruplarına verilen ad

7.3.*(10p)* Yönlü bir grafa ait düğümler ve kenar bilgisi şu şekilde tutulmaktadır: **dugum1**, **dugum2**, **agirlik** Veri dosyasındaki örnek satırların listesi aşağıdaki gibi olsun.

1, 2, 2
1, 3, 5
1, 4, 4
2, 4, 1
3, 5, 2

5p



Örnek veri için grafı çizerek her bir düğüme gelen kenarların sayısını (in degree) aşağıdaki gibi çıktı verecek şekilde (düğüm, sayi) herhangi bir programlama dilinde sözde kod şeklinde veriniz.

2, 1
3, 1
4, 2
5, 1

map-reduce mantığında çözebiliriz:

```
map(k, v) {
  k' = v.split(',')[1]  // ilanci degeri
  v' = 1
  emit(k', v') }

reduce(k, v[]) {
  k' = k
  toplam = 0
  for each x in v[]
    toplam += x
  v' = toplam
  emit(k', v') }
```

veya Cypher dilinde yazabiliriz.

```
MATCH (u:Node)
RETURN u.name, size((u)<--()) as gelen
```

8. (Metin Analizi)

8.1. *(5p)* Yapısal, yarı-yapısal ve yapısal olmayan veri nedir?

Yapısal veri = modellenmesi, saklanması, sorgulanması, işlenmesi kolay; belirli boyutlarda önceden tanımlı alanlara sahip veri: RDBMS tablolarındaki gibi. Yapısal olmayan = tanımlı bir format haricinde depolanan veri: e-posta, resim vs. Yarı-yapısal: meta-modelleri barındıran veri: XML, JSON gibi.

8.2. *(5p)* Metin madenciliği adımlarının/pipeline yapısının isimlerini veriniz?

metin ön işleme → metin dönüşümü → özellik seçimi → veri madenciliği → Değerlendirme

Dr. Süleyman Eken

Başarılar dilerim.

100201068	E
120201099	50
120201571	E
130201021	11
130201028	E
130201046	E
130201105	34
140201024	70
140201026	E
140201033	40
140201089	42
140201109	82
140201115	75
140201135	57
150201101	67
150201111	82
150201113	72
150201114	E
150201120	58
150201123	E
150201137	E
150201140	66
150201141	91
150201143	20
150201161	50
150201167	71
150201169	62
150201170	83
***150201172**	110
150201176	E
150201178	76
150201197	13
150201198	76
150201200	78
150202113	E
160201117	74
160201118	68
170201098	77
170201100	62

*10 puan vizeye eklenecektir

120202017	43
130202014	37
130202026	17
130202040	E
130202096	E
130202117	E
140202036	E
140202051	23
150202008	76
150202009	83
150202010	74
150202011	75
150202013	67
150202014	47
150202030	5
150202033	E
150202041	79
150202049	83
150202054	70
150202056	19
150202058	70
150202061	82
150202068	79
150202079	93
150202082	78
150202084	67
150202085	63
150202086	72
150202089	75
150202097	61
150202110	79
150202114	E
150202142	E
160202090	E
170202113	E
170202117	E

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

1. ogr istatistiksel sonuclari

In [3]:

```
df = pd.read_csv("/home/ipcvlab/Downloads/1ogr.csv")
```

In [4]:

```
df.iloc[:,-1].describe()
```

Out[4]:

```
count     29.000000
mean      62.655172
std       22.869624
min       11.000000
25%       50.000000
50%       68.000000
75%       76.000000
max      110.000000
Name: Not, dtype: float64
```

In [5]:

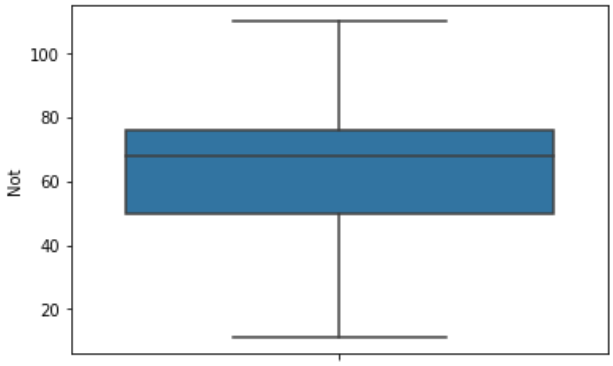
```
df = df.dropna(subset=['Not'])
```

In [6]:

```
sns.boxplot(y=df["Not"])
```

Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa6e84b5850>
```

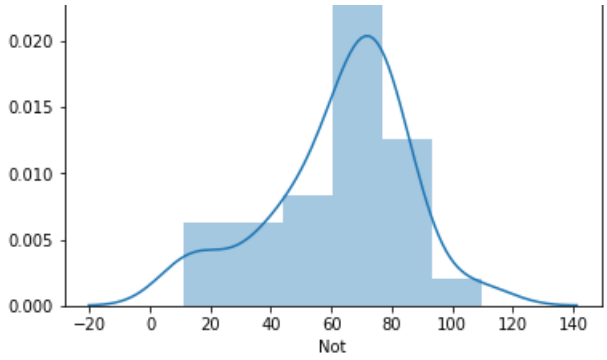


In [7]:

```
sns.distplot(df['Not'])
```

Out[7]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa70ff946d0>
```



1. ogr soru bazında detaylı istatistiksel sonuclar

In [8]:

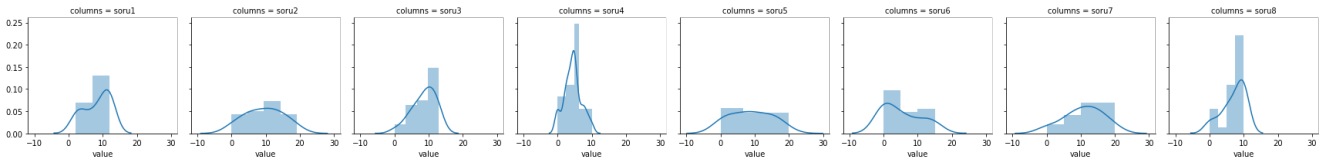
```
df2 = pd.read_csv("/home/ipcvlab/Downloads/1ogrAyrinti.csv")
df2 = df2.iloc[:, 1:9]
df2 = df2.dropna()
df2.describe()
```

Out[8]:

	soru1	soru2	soru3	soru4	soru5	soru6	soru7	soru8
count	29.000000	29.000000	29.000000	29.000000	29.000000	29.000000	29.000000	29.000000
mean	8.241379	9.482759	8.586207	4.448276	8.793103	5.275862	10.793103	7.034483
std	3.851300	5.558910	3.459121	2.543707	6.084989	5.618188	5.665556	3.396245
min	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	4.000000	6.000000	6.000000	3.000000	4.000000	0.000000	7.000000	5.000000
50%	10.000000	10.000000	9.000000	5.000000	9.000000	5.000000	10.000000	8.000000
75%	12.000000	14.000000	11.000000	5.000000	15.000000	10.000000	15.000000	10.000000
max	12.000000	19.000000	13.000000	10.000000	20.000000	15.000000	20.000000	10.000000

In [9]:

```
dfm = df2.melt(var_name='columns')
g = sns.FacetGrid(dfm, col='columns')
g = (g.map(sns.distplot, 'value'))
```



2. ogr istatistiksel sonuclari

In [10]:

```
df3 = pd.read_csv("/home/ipcvlab/Downloads/2ogr.csv")
df3.iloc[:,-1].describe()
```

Out[10]:

```
count    26.000000
mean     62.192308
std      23.814314
min       5.000000
25%      50.500000
50%      71.000000
75%      78.750000
max      93.000000
```

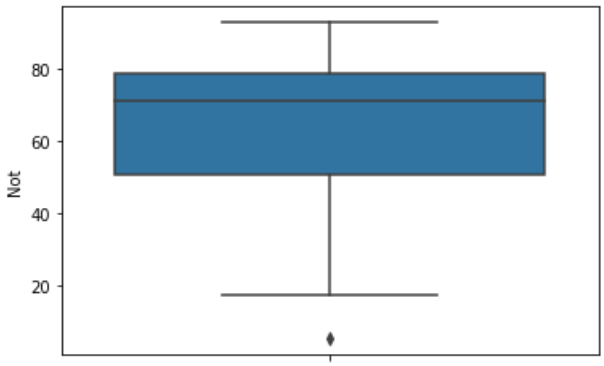
```
Name: Not, dtype: float64
```

In [11]:

```
df3 = df3.dropna(subset=['Not'])
sns.boxplot(y=df3["Not"])
```

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa70ea76b50>
```

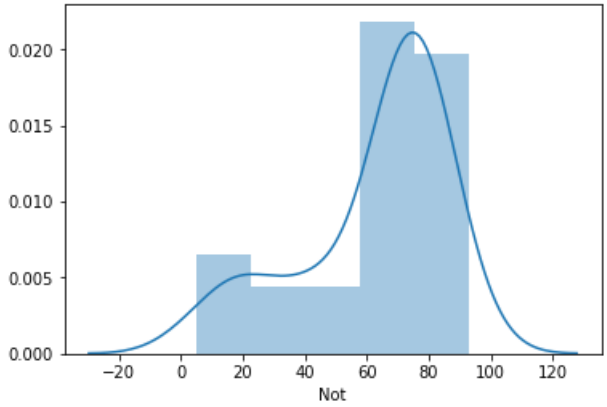


In [12]:

```
sns.distplot(df3['Not'])
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa70e93ef50>
```



2. ogr soru bazında detaylı istatistiksel sonuclar

In [14]:

```
df4 = pd.read_csv("/home/ipcvlab/Downloads/2ogrAyrinti.csv")
df4 = df4.iloc[:, 1:9]
df4 = df4.dropna()
df4.describe()
```

Out[14]:

	soru1	soru2	soru3	soru4	soru5	soru6	soru7	soru8
count	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000	26.000000
mean	7.923077	8.692308	9.076923	3.500000	10.538462	5.076923	11.307692	6.076923
std	4.371938	4.611024	3.654291	2.915476	6.236863	3.719388	5.746437	3.938762
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	4.250000	6.000000	6.250000	0.250000	5.000000	2.000000	8.250000	3.000000
50%	10.000000	7.500000	11.000000	3.000000	11.500000	4.500000	12.500000	7.000000

	soru1	soru2	soru3	soru4	soru5	soru6	soru7	soru8
50%	10.000000	7.500000	11.000000	3.000000	11.500000	4.500000	12.500000	7.000000
75%	12.000000	12.000000	12.000000	6.000000	15.000000	8.000000	15.000000	10.000000
max	12.000000	17.000000	13.000000	10.000000	20.000000	13.000000	20.000000	10.000000

In [15]:

```
dfm2 = df4.melt(var_name='columns')
g = sns.FacetGrid(dfm2, col='columns')
g = (g.map(sns.distplot, 'value'))
```