Python ile Veri Bilimine Giriş Dersi Final Projesi

Ad: İrem

Soyad: Aksoy

Numara:201001001

Veri Seti Hikayesi

https://www.kaggle.com/datasets/spscientist/students-performance-in-exams

Veriseti bir grup öğrencinin sahip olduğu koşulları ve sınavlardan almış oldukları sonuçların bilgisini içermektedir. Verisetinde şu bilgiler yer almaktadır:

- gender (male, female)
- race/ethnicity (A,B,C,D,E)
- parental level of education (some college, associate's degree, high school, some high school, bachelor's degree,master degree)
- lunch (standard, free/reduced)
- test preparation course (none, completed)
- math score (0-100)
- reading score (0-100)
- writing score (0-100)

1.Veri Seti Hakkında Genel Bilgiler

In [53]:
```
# Öncelikle dosyadan veri okuma işlemi yapılır.
import pandas as pd
studentPerformance =pd.read_csv("archive/StudentsPerformance.csv")
studentPerformance
```

Out[53]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

1000 rows × 8 columns

In [54]:
```
df = studentPerformance.copy() # Verisetinin kopyası oluşturuldu.
df.info() # Bu kod ile birlikte veriseti hakkında genel bilgi edinilir.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   gender                       1000 non-null   object
 1   race/ethnicity               1000 non-null   object
 2   parental level of education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test preparation course      1000 non-null   object
 5   math score                   1000 non-null   int64 
 6   reading score                1000 non-null   int64 
 7   writing score                1000 non-null   int64 
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

In [55]:
```
df.head() # İlk beş satırın görüntülenmesini sağlandı.
```

Out[55]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

In [56]:
```
df.tail() # Son beş satırın görüntülenmesi sağlandı.
```

Out[56]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

1.1. Object değişkenini kategorik değişkene çevrimi

In [57]:
```
df.rename(columns={"race/ethnicity":"ethnicity"}, inplace=True) # Burada sütun isimleri, uygun isimler ile değiş
```

In [58]:
```
df.rename(columns={"parental level of education":"parental_level_of_education"} , inplace=True)
```

In [59]:
```
df.rename(columns={"test preparation course":"test_preparation_course"},inplace=True)
```

In [60]:
```
df.rename(columns={"math score":"math_score"} ,inplace=True)
```

In [61]:
```
df.rename(columns={"reading score":"reading_score"} ,inplace=True)
```

In [62]:
```
df.rename(columns={"writing score":"writing_score"}, inplace=True)
```

In [63]:
```
df.head() # sütun isimlerindeki değişiklikler gözlemlendi.
```

Out[63]:

	gender	ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score	reading_score	writing_score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

In [64]:
```
df.gender = pd.Categorical(df.gender) # Object değişkenler kategorik değişkenlere çevrildi.
df.ethnicity = pd.Categorical(df.ethnicity)
df.parental_level_of_education = pd.Categorical(df.parental_level_of_education)
df.lunch = pd.Categorical(df.lunch)
df.test_preparation_course = pd.Categorical(df.test_preparation_course)
```

In [65]:
```
df.info() # Üstte yapılan işlemin sonucu gözlemlendi.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype
```

```
 ---  ------                       --------------  -----
  0   gender                       1000 non-null   category
  1   ethnicity                    1000 non-null   category
  2   parental_level_of_education  1000 non-null   category
  3   lunch                        1000 non-null   category
  4   test_preparation_course      1000 non-null   category
  5   math_score                   1000 non-null   int64
  6   reading_score                1000 non-null   int64
  7   writing_score                1000 non-null   int64
dtypes: category(5), int64(3)
memory usage: 29.2 KB
```

2.Veri Seti ile İlgili istatistikler

In [23]:
```
df.describe() # Burada veriseti ile ilgili genel bir istatistik çıkarılır. Sonuç olarak öğrencilerin bazı dersle
#en yüksek ve en düşük notları, notların standart sapmaları, notların genel ortalaması gibi değerler incelenebil
```

Out[23]:

	math_score	reading_score	writing_score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

In [24]:
```
df.describe().T # Burada verisetinin transpose'u alınarak veri,daha okunaklı bir hale getirilmektedir.
```

Out[24]:

	count	mean	std	min	25%	50%	75%	max
math_score	1000.0	66.089	15.163080	0.0	57.00	66.0	77.0	100.0
reading_score	1000.0	69.169	14.600192	17.0	59.00	70.0	79.0	100.0
writing_score	1000.0	68.054	15.195657	10.0	57.75	69.0	79.0	100.0

3.Eksik değerlerin gözlemlenmesi

In [25]:
```
df.isnull().values.any() # Burada eksik değerlerin varlığı gözlemlenmiştir. Sonuç olarak eksik bir değerin olmadı
```

Out[25]:
```
False
```

4. Kategorik ve Sürekli Değişkenlerin İncelenmesi

In [19]:
```
katdf = df.select_dtypes(include = ["category"]) # Veriseti içinden tipi kategorik olan değişkenler seçildi.
```

In [20]:
```
katdf.head()
```

Out[20]:

	gender	ethnicity	parental_level_of_education	lunch	test_preparation_course
0	female	group B	bachelor's degree	standard	none
1	female	group C	some college	standard	completed
2	female	group B	master's degree	standard	none
3	male	group A	associate's degree	free/reduced	none
4	male	group C	some college	standard	none

In [21]:
```
katdf.gender.unique() # cinsiyette kategorilerin neler olduğu gözlemlendi.
```

Out[21]:
```
['female', 'male']
Categories (2, object): ['female', 'male']
```

In [22]:
```
katdf["gender"].value_counts() # Veri setindeki cinsiyet dağılımı
```

Out[22]:
```
female    518
male      482
Name: gender, dtype: int64
```

In [23]:
```
katdf.ethnicity.unique()
```

Out[23]:
```
['group B', 'group C', 'group A', 'group D', 'group E']
Categories (5, object): ['group A', 'group B', 'group C', 'group D', 'group E']
```

In [24]:
```
katdf["ethnicity"].value_counts()
```

Out[24]:
```
group C    319
group D    262
group B    190
group E    140
group A     89
Name: ethnicity, dtype: int64
```

In [25]:
```
katdf.parental_level_of_education.unique()
```

Out[25]:
```
['bachelor's degree', 'some college', 'master's degree', 'associate's degree', 'high school', 'some high school']
Categories (6, object): ['associate's degree', 'bachelor's degree', 'high school', 'master's degree', 'some colle
ge', 'some high school']
```

In [26]:
```
katdf["parental_level_of_education"].value_counts()
```

Out[26]:
```
some college          226
associate's degree    222
high school           196
some high school      179
bachelor's degree     118
master's degree        59
Name: parental_level_of_education, dtype: int64
```

In [27]:
```
katdf.lunch.unique()
```

Out[27]:
```
['standard', 'free/reduced']
Categories (2, object): ['free/reduced', 'standard']
```

In [28]:
```
katdf["lunch"].value_counts().count() #Öğle yemeğinde 2 kategori olduğu sonucuna varıldı.
```

Out[28]:
```
2
```

In [29]:
```
katdf["lunch"].value_counts() # Bu iki kategorinin veri seti içinde nasıl dağıldığı gözlemlendi.
```

Out[29]:
```
standard        645
free/reduced    355
Name: lunch, dtype: int64
```

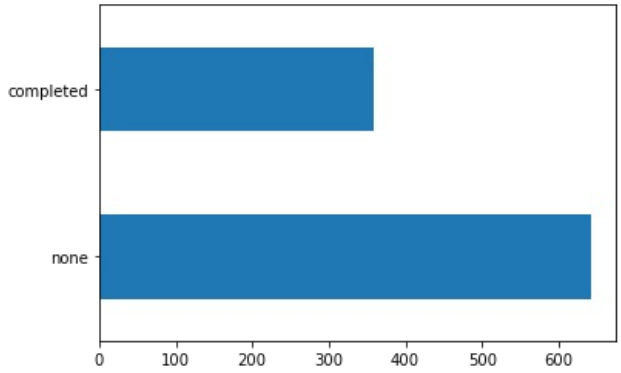
In [30]:
```
katdf.test_preparation_course.unique()
```

Out[30]:
```
['none', 'completed']
Categories (2, object): ['completed', 'none']
```

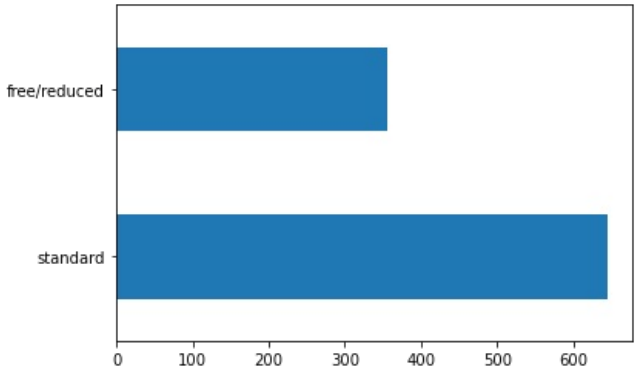
In [31]:
```
katdf["test_preparation_course"].value_counts()
```

Out[31]:
```
none         642
completed    358
Name: test_preparation_course, dtype: int64
```

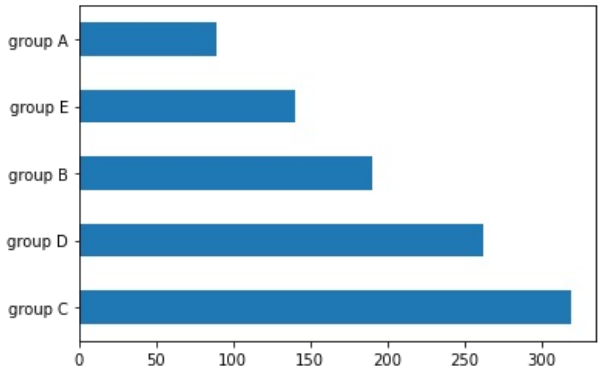
In [32]:
```
df["test_preparation_course"].value_counts().plot.barh(); # Üstte edinilen bilgiler tablo hsline getirildi.
```



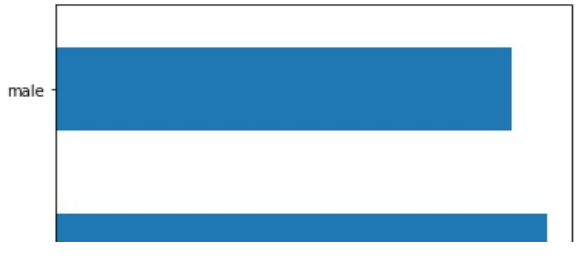
In [14]:
```
df["lunch"].value_counts().plot.barh();
```

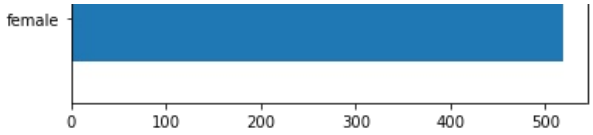


In [15]:
```
df["ethnicity"].value_counts().plot.barh();
```

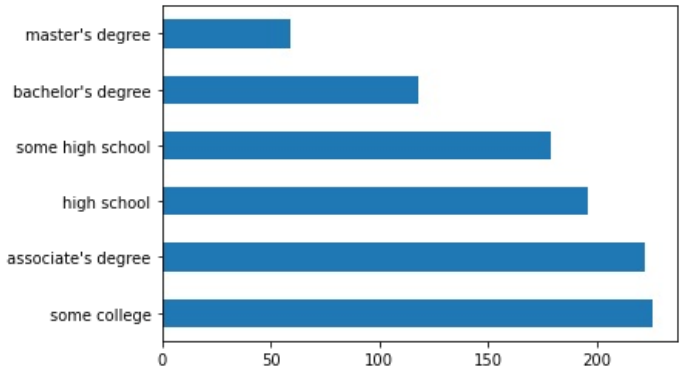


In [33]:
```
df["gender"].value_counts().plot.barh();
```





In [34]:
```
df["parental_level_of_education"].value_counts().plot.barh();
```



In [35]:
```
# Sürekli değişkenlerin incelenmesi
numdf = df.select_dtypes(include = [ "int64"])
```

In [36]:
```
numdf.head() # Sayısal değişkenlerin verisetindeki ilk beş değeri gözlemlendi.
```

Out[36]:

	math_score	reading_score	writing_score
0	72	72	74
1	69	90	88
2	90	95	93
3	47	57	44
4	76	78	75

In [37]:
```
numdf.describe().T # sürekli değişkenlerin istatiksel incelenmesi
```

Out[37]:

	count	mean	std	min	25%	50%	75%	max
math_score	1000.0	66.089	15.163080	0.0	57.00	66.0	77.0	100.0
reading_score	1000.0	69.169	14.600192	17.0	59.00	70.0	79.0	100.0
writing_score	1000.0	68.054	15.195657	10.0	57.75	69.0	79.0	100.0

5. Veriseti Üzerinde Yapılan İncelemeler

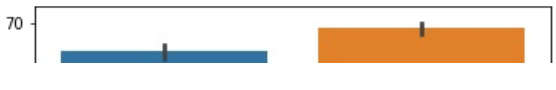
5.1.Cinsiyetin sınav notları üzerindeki etkisi

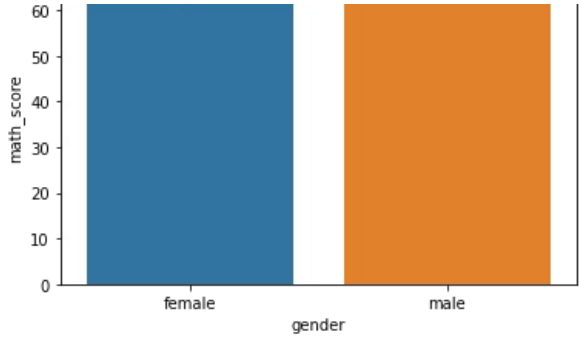
In [38]:
```
df.pivot_table("math_score", index = "gender") # Cinsiyete göre kadın ve erkek öğrencilerin not ortalması gözlemi
```

Out[38]:

gender	math_score
female	63.633205
male	68.728216

In [39]:
```
import seaborn as sns # sonuç olarak erkeklerin matematik ortalamasının daha yüksek olduğu gözlemdi
sns.barplot(x = "gender", y = "math_score",data = df); #Sütun grafiği olarak incelemek için barplot kullanıldı.
```



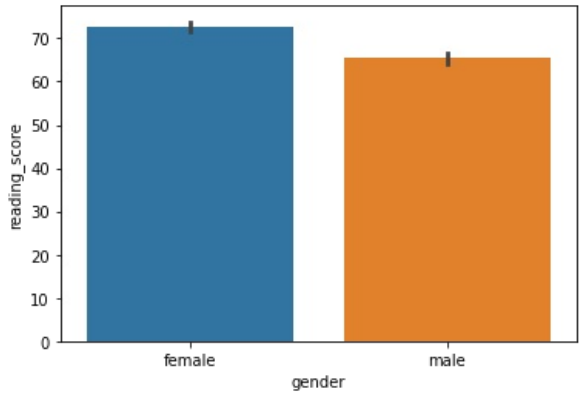


In [60]:
```
df.pivot_table("reading_score", index = "gender") # Reading dersi sonuçlarının cinsiyete göre ortalaması incelend
```

Out[60]:

gender	reading_score
female	72.608108
male	65.473029

In [61]:
```
sns.barplot(x = "gender", y = "reading_score",data = df);  # Sonuç olarak kadın öğrencilerin daha yüksek bir orta
# olduğu görüldü.
```

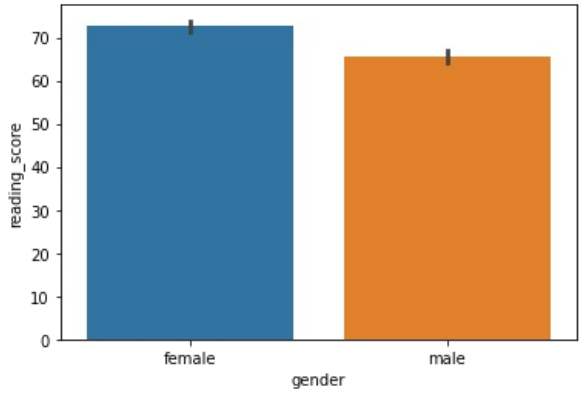


In [62]:
```
df.pivot_table("writing_score", index = "gender") # Writing dersinde sonuçlar cinsiyete göre incelendi.
```

Out[62]:

gender	writing_score
female	72.467181
male	63.311203

In [63]:
```
sns.barplot(x = "gender", y = "reading_score",data = df); # Sütun grafiği tablosu ile ortalama olarak kadın öğren
# başarılı olduğu gözlemlendi.
```



5.2 Cinsiyetin ve ebeveynlerin eğitim düzeyinin, sınav notları üzerindeki etkisi

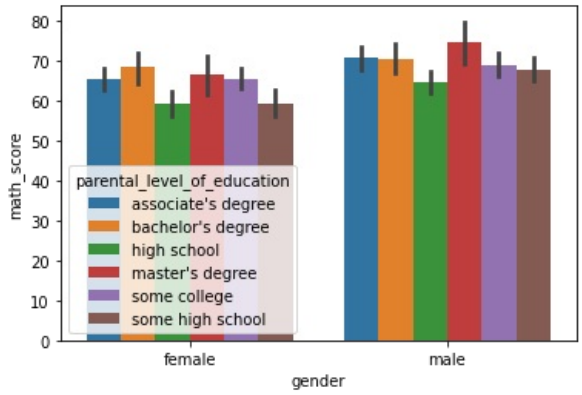
5.2.Cinsiyetin ve ebeveynlerin eğitim düzeyinin, sınav notları üzerindeki etkisi

In [64]:
```
df.pivot_table("math_score", index = "gender", columns = "parental_level_of_education") # Cinsiyetin ve ebeveynle
#durumunun matemetik dersi üzerine etkisini gösteren kod yazıldı.
```

Out[64]:

parental_level_of_education	associate's degree	bachelor's degree	high school	master's degree	some college	some high school
gender						
female	65.250000	68.349206	59.351064	66.500000	65.406780	59.296703
male	70.764151	70.581818	64.705882	74.826087	69.009259	67.840909

In [66]:
```
sns.barplot(x = "gender", y = "math_score", hue = "parental_level_of_education", data = df);
# Sonuç olarak en yüksek matematik ortlamasına sahip olan kadın öğrencilerin ebeveynlerinin lisans derecesine, er
# öğrencilerinin ebeveylerinin ise master derecesine sahip olduğu gözlemlendi.
```

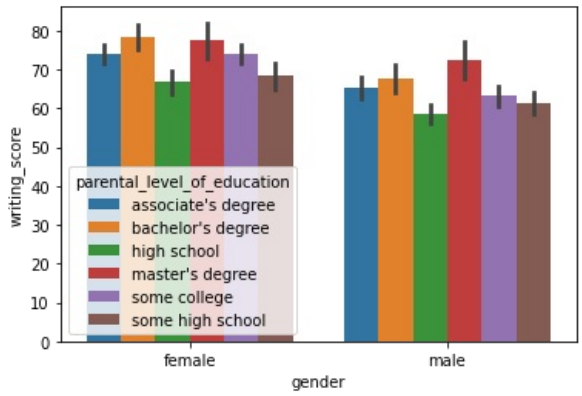


In [67]:
```
df.pivot_table("writing_score", index = "gender", columns = "parental_level_of_education") # Aynı durum için writ
# incelemesi yapıldı.
```

Out[67]:

parental_level_of_education	associate's degree	bachelor's degree	high school	master's degree	some college	some high school
gender						
female	74.00000	78.380952	66.691489	77.638889	74.050847	68.285714
male	65.40566	67.654545	58.539216	72.608696	63.148148	61.375000

In [68]:
```
sns.barplot(x = "gender", y = "writing_score", hue = "parental_level_of_education", data = df);
# Burada da sonuç kadın öğrencilerin not oratlaması daha yüksek olsa da ebeveynlerin eğitim düzeylerinin notlara
# ile aynı olduğu gözlemlendi.
```

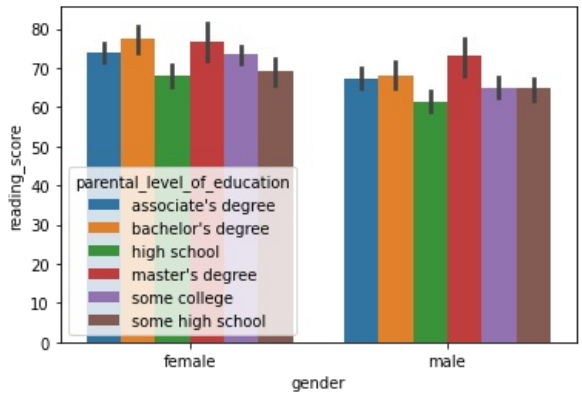


In [69]:
```
df.pivot_table("reading_score", index = "gender", columns = "parental_level_of_education") # Burada ise reading c
# yukarıdaki inceleme yapıldı.
```

Out[69]:

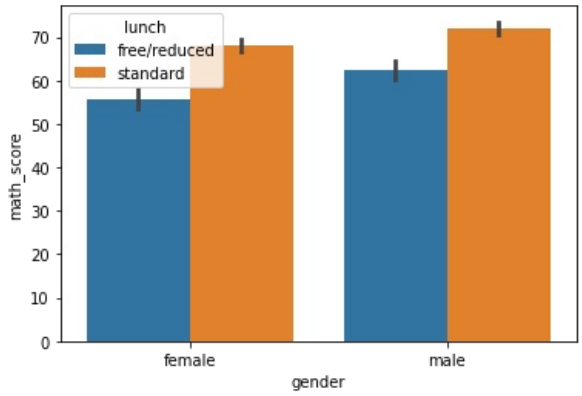
parental_level_of_education	associate's degree	bachelor's degree	high school	master's degree	some college	some high school
gender						
female	74.120690	77.285714	68.202128	76.805556	73.550847	69.109890
male	67.433962	68.090909	61.480392	73.130435	64.990741	64.693182

In [70]:
```
sns.barplot(x = "gender", y = "reading_score", hue = "parental_level_of_education", data = df);
# Burada da somuç olarak bir değişiklik gözlemlenmedi.
```

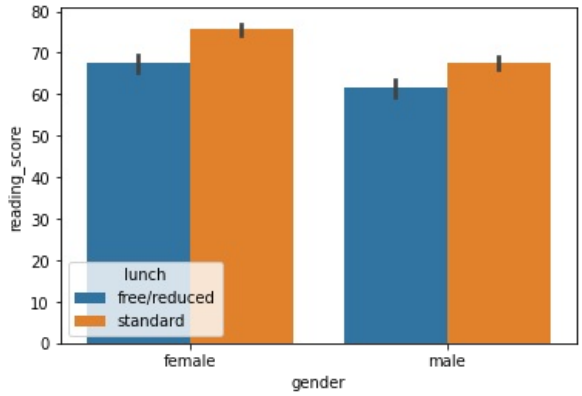


5.3. Cinsiyetin ve Yemek Seçiminin Notlar Üzerindeki Etkisi

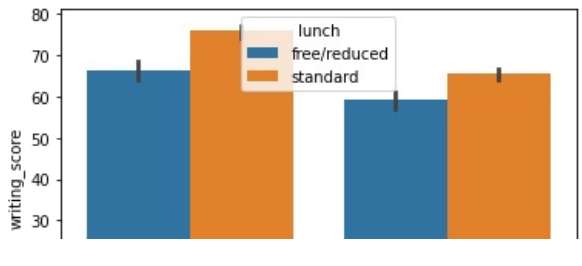
In [71]:
```
sns.barplot(x = "gender", y = "math_score", hue = "lunch", data = df); # genel olarak standart yemek yiyenlerin d
# aldıkları gözlemlendi
```

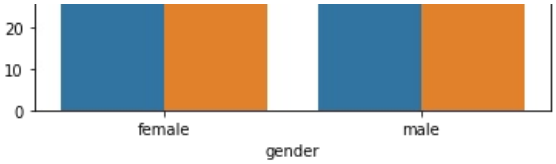


In [72]:
```
sns.barplot(x = "gender", y = "reading_score", hue = "lunch", data = df); # Burada kadın olan  ve standart yemek
# ortalmasının daha yüksek olduğu gözlemlendi.
```



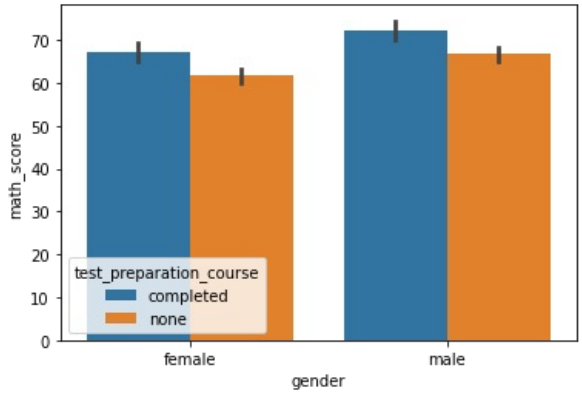
In [73]:
```
sns.barplot(x = "gender", y = "writing_score", hue = "lunch", data = df); # Burada kadın öğrencilerin yemek terci
# dahi olsa erkek öğrencilerden daha yüksek olduğu gözlemlendi.
```



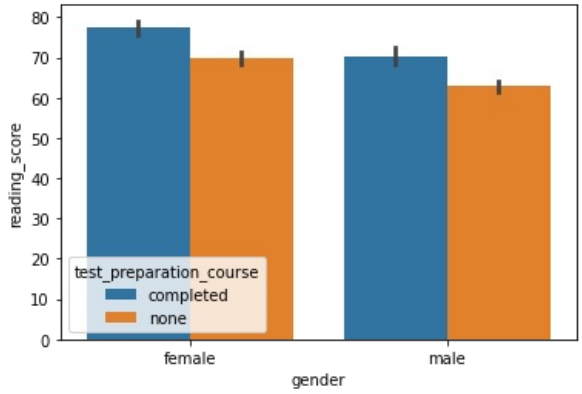


5.4 Cinsiyetin ve kursa gitmenin notlar üzerindeki etkisi

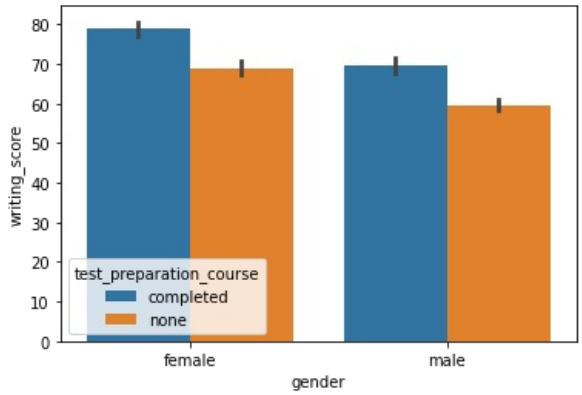
In [74]:
```
sns.barplot(x = "gender", y = "math_score", hue = "test_preparation_course", data = df);
# Kurs tamamlamanın matematik dersinde ortalamaya olumlu bir etkisi olduğu gözlemlendi.
```



In [75]:
```
sns.barplot(x = "gender", y = "reading_score", hue = "test_preparation_course", data = df);
# Burada reading dersinde, kurs tamamlamanın kadın ve erkek öğrencilerde olumlu bir etkiye sahip oldupu gözlemler
```



In [76]:
```
sns.barplot(x = "gender", y = "writing_score", hue = "test_preparation_course", data = df);
# Burada writing dersinde, kurs tamamlamanın kadın ve erkek öğrencilerde olumlu bir etkiye sahip oldupu gözlemler
```

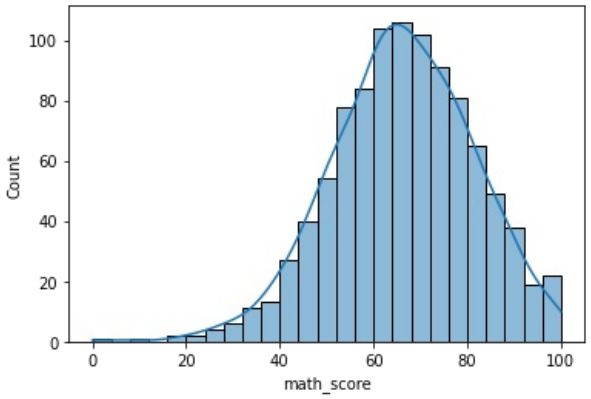


5.5 Verilerin histogram grafiği ile incelenmesi

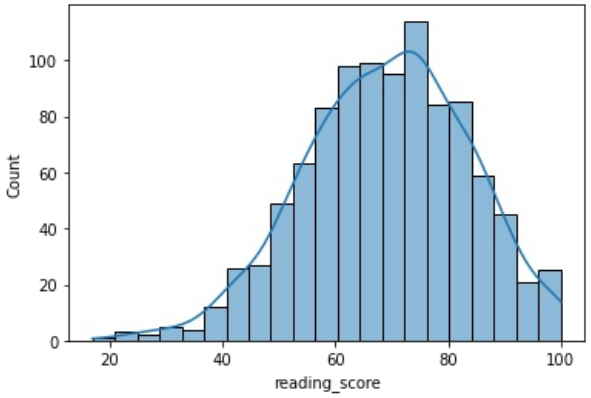
Histogram grafikleri veri dağılımını gözlemlemek için kullanılmaktadır.

In [42]:

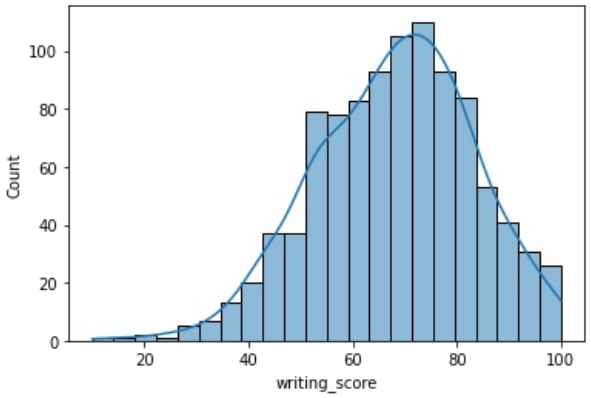
In [42]:
```
sns.histplot(df.math_score,kde=True); # Burada matematik dersinde notların 60 ile 80 arasında yoğunlaştığı görülm
```



In [46]:
```
sns.histplot(df.reading_score,kde=True); # Bu derste de yogunluğun aynı değerler arasında olduğu görülmektedir.
```



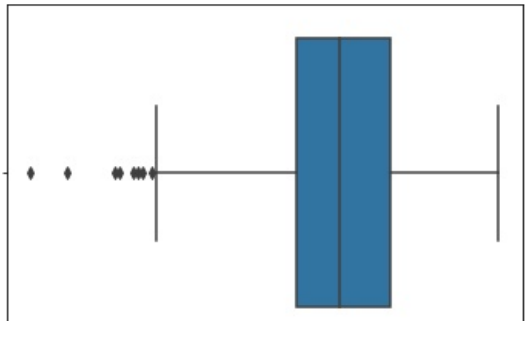
In [47]:
```
sns.histplot(df.writing_score,kde=True); # writing dersinde notların genel olarak 60 ile 80 arasında yoğunlaştığı
```

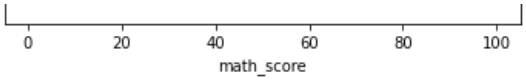


5.6. Verilerin Boxplot Grafiği ile İncelenmesi

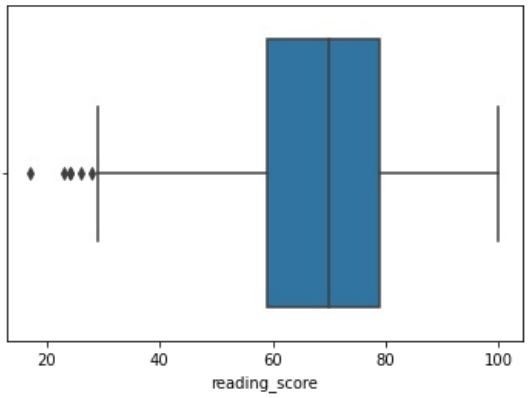
Boxplot grafikleri veri görselleştirmenin yanı sıra veriler hakkında istatiksel özellikleri görselleştirmek için de kullanılır.

In [48]:
```
sns.boxplot(x = df["math_score"]); # matematikte min değerin 0,max değerin 100, medyanın ise 60 ve 80 arasında bi
# görülmektedir.
```

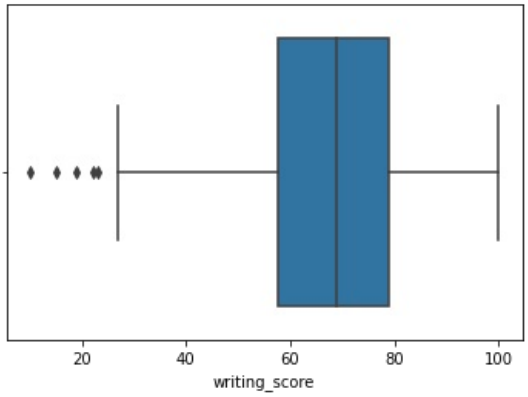




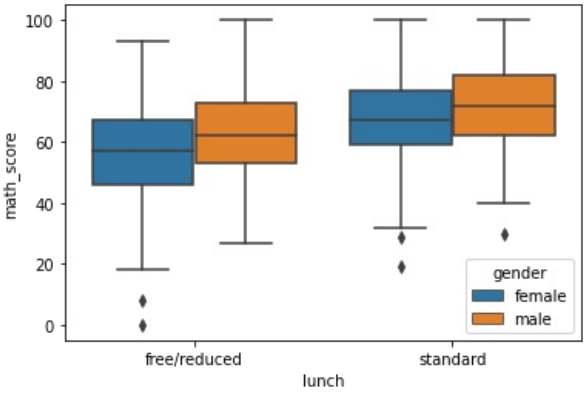
In [36]:
```
sns.boxplot(x = df["reading_score"]); # reading dersinde min değerinin 20 ye yakın bir değer olduğu, max değerini
# ve ortalamanın 60 ile 80 arasında bir değer olduğu görülmektedir.
```



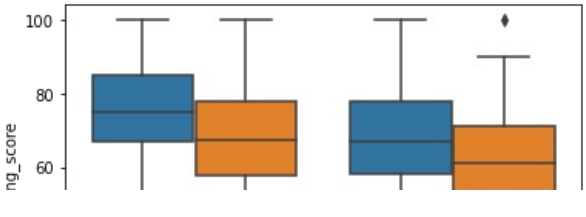
In [37]:
```
sns.boxplot(x = df["writing_score"]); # writing dersisnde min değerin 20 nin altında bir değer olduğu, max değeri
# ortalamanın ise 60 ile 80 arasında bir değer olduğu görülmektedir.
```

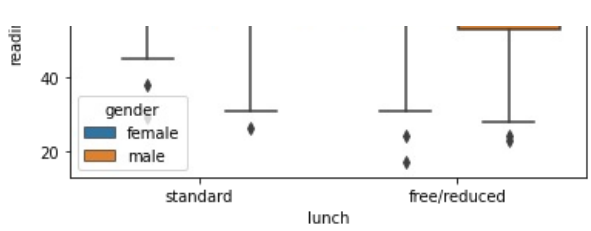


In [49]:
```
sns.boxplot(x = "lunch", y = "math_score", hue = "gender", data = df); # Burada matemetik dersi için standart yem
# sınavda daha olumlu bir etkisi olduğu görülmüştür ve genel olarak bu ders için erkeklerin kadınlardan daha başa
# görülmüştür.Aynı zamanda 0 alan kişinin kadın öğrenci olduğu ve indirimli yemek tercih ettiği anlaşılmıştır.
```



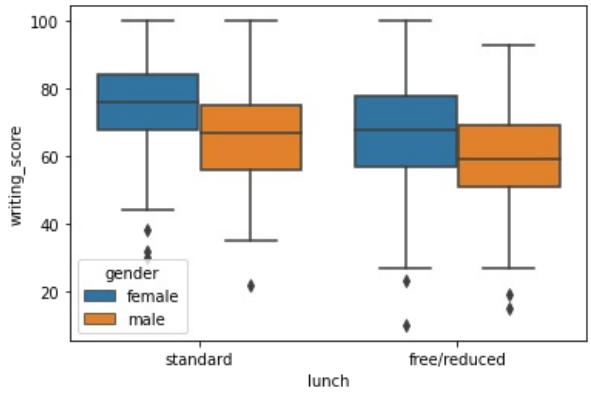
In [60]:
```
sns.boxplot(x = "lunch", y = "reading_score", hue = "gender", data = df); # Reading dersinde de en düşük notları
# lerin aldığı ve yemek olarak indirimli yemek tercih ettiği görülmektedir.Ayrıca standart yemek tercih eden kadın
# en yüksek not ortalamasına sahip olduğu görülmektedir.
```





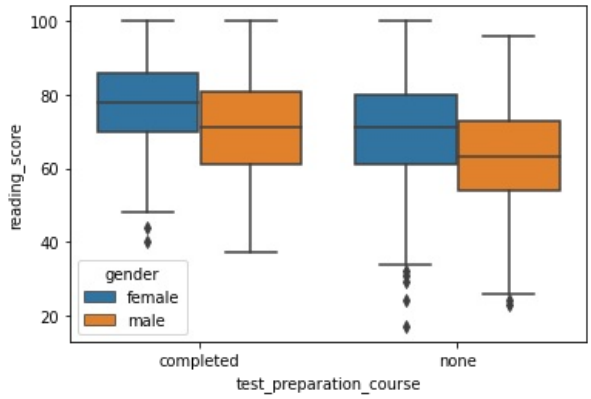
In [61]:

```
sns.boxplot(x = "lunch", y = "writing_score", hue = "gender", data = df); # writing dersinde de en yüksek not ort
# standart yemek tercih eden kadın öğrenciler olduğu görülmektedir.Ayrıca yine en düşük notları indirimli yemek t
# kadın öğrenciler almıştır.
```



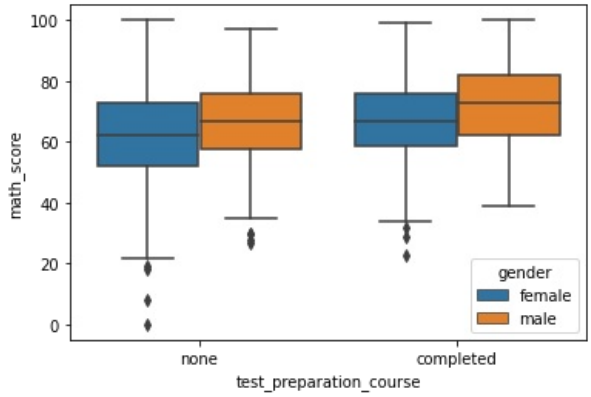
In [52]:

```
sns.boxplot(x = "test_preparation_course", y = "reading_score", hue = "gender", data = df);# Bu grafikte hazırlık
#tamamlamanın sınav sonucuna olumlu bir etkisi olduğu görülmştür ve sınavda genel olarak kadınlar erkeklerden dah
# ğu sonucuna varılmıştır. Aynı zamanda en düşük notların kadın öğrenciler tarafından aldığı gözlemlenmiştir.
```



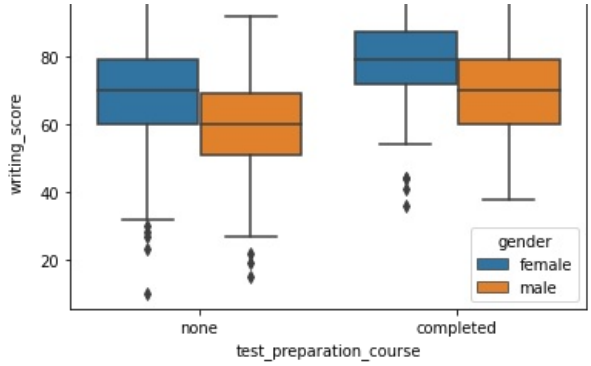
In [62]:

```
sns.boxplot(x = "test_preparation_course", y = "math_score", hue = "gender", data = df); # Matematik dersinde de
# erkek öğrencilerin en yüksek notlara sahip olduğu görülmektedir. En düşük notları ise kurs tamamlamayan kadın ö
# tarafından aldığı görülmektedir.
```



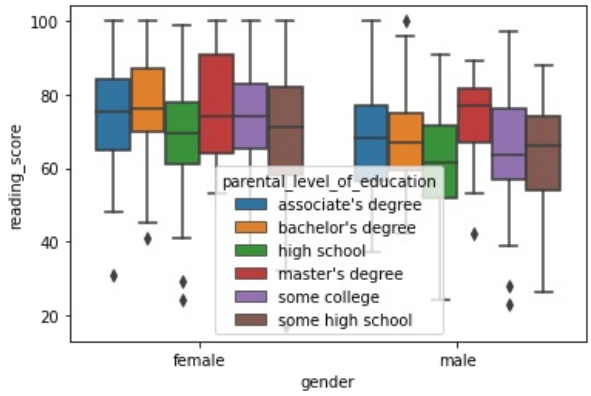
In [63]:

```
sns.boxplot(x = "test_preparation_course", y = "writing_score", hue = "gender", data = df); #writing dersinde de
# öğrencilerin not ortalaması daha yüksektir.Ayrıca endüşük notu kursu tamamlamayan kadın öğrencilerin aldığı gör
```



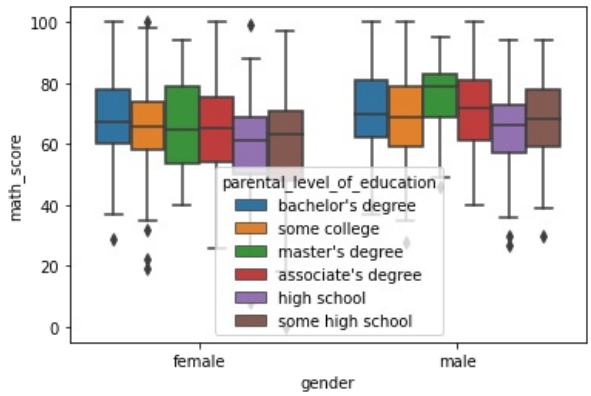
In [54]:

```
sns.boxplot(x = "gender", y = "reading_score", hue = "parental_level_of_education", data = df); #reading dersinde
#ebeveylerinin lisans derecesine sahip olanların daha yüksek ortalamaya sahip olduğu görülmüştür. Erkeklerde ise
# derecesine sahip olanlar yüksek not ortalamasına sahiptir.
```



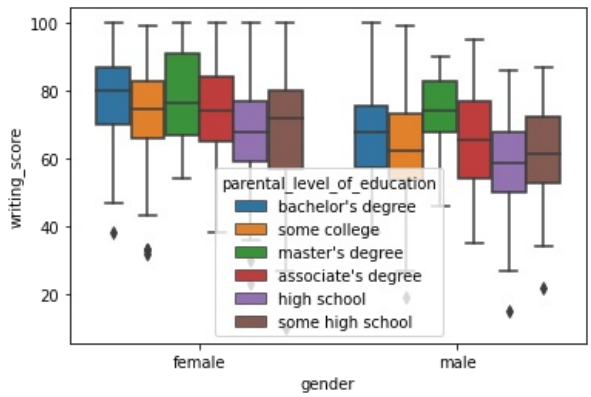
In [58]:

```
sns.boxplot(x = "gender", y = "math_score", hue = "parental_level_of_education", data = df); # matematik dersinde
#ortalamasını erkek öğrencilerde yüksek olduğu ve ebeveylerinin master derecesine sahip olduğu görülmektedir.
```

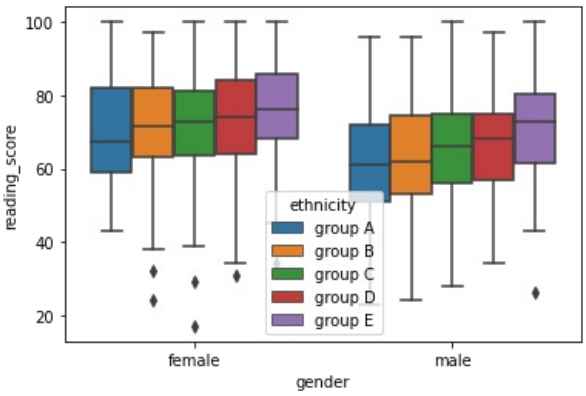


In [59]:

```
sns.boxplot(x = "gender", y = "writing_score", hue = "parental_level_of_education", data = df); # writing dersinc
#kadın öğrencilerin yüksek not ortlamasına sahip olduğu görülmektedir. Ayrıca kadın öğrencilerde ebeveynlerinin
# fark etmeksizin yüksek not alabildikleri görülmektedir.
```



In [56]:
```
sns.boxplot(x = "gender", y = "reading_score", hue = "ethnicity", data = df);#Bu grafikte kadınlarda etnik grubun
# etki etmediğini sadece A grubunda olanların ortalmasının daha düşük olduğu görülmektedir. Erkeklerde ise ortala
# da daha yüksek olduğu görülmektedir
```



6.Veriseti Üzerinde Betimsel İstatistik

Betimsel istatistik için researchpy kütüphanesi import edilmelidir. researchpy kütüphanesi araştırma ve istatistiksel analizlerde kullanılan bazı yaygın işlemleri kolaylaştırmak için tasarlanmıştır.

In [15]:
```
!pip install researchpy
import researchpy as rp
```

```
Requirement already satisfied: researchpy in c:\anaconda3\lib\site-packages (0.3.5)
Requirement already satisfied: numpy in c:\anaconda3\lib\site-packages (from researchpy) (1.20.3)
Requirement already satisfied: pandas in c:\anaconda3\lib\site-packages (from researchpy) (1.3.4)
Requirement already satisfied: scipy in c:\anaconda3\lib\site-packages (from researchpy) (1.7.1)
Requirement already satisfied: patsy in c:\anaconda3\lib\site-packages (from researchpy) (0.5.2)
Requirement already satisfied: statsmodels in c:\anaconda3\lib\site-packages (from researchpy) (0.12.2)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\anaconda3\lib\site-packages (from pandas->researchpy)
(2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\anaconda3\lib\site-packages (from pandas->researchpy) (2021.3)
Requirement already satisfied: six>=1.5 in c:\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas->r
esearchpy) (1.16.0)
```

In [26]:
```
rp.summary_cont(df[["math_score","reading_score","writing_score"]]) # Sayısal değişkenlerle ilgili özet bir istat
```

Out[26]:

	Variable	N	Mean	SD	SE	95% Conf.	Interval
0	math_score	1000.0	66.089	15.1631	0.4795	65.1481	67.0299
1	reading_score	1000.0	69.169	14.6002	0.4617	68.2630	70.0750
2	writing_score	1000.0	68.054	15.1957	0.4805	67.1110	68.9970

In [ ]:
```
# Bu tabloda kayıt sayısı, ortalama, standart sapma,standart hata, değişkenlerin %95 güven aralığı ve güven aralı
# üst sınırlarının bilgisi verilmektedir.
```

In [28]:
```
rp.summary_cat(df[["gender","ethnicity","parental_level_of_education","lunch","test_preparation_course"]])
```

Out[28]:

	Variable	Outcome	Count	Percent
0	gender	female	518	51.8
1		male	482	48.2
2	ethnicity	group C	319	31.9
3		group D	262	26.2
4		group B	190	19.0
5		group E	140	14.0
6		group A	89	8.9
7	parental_level_of_education	some college	226	22.6
8		associate's degree	222	22.2
9		high school	196	19.6

10		some high school	179	17.9
11		bachelor's degree	118	11.8
12		master's degree	59	5.9
13	lunch	standard	645	64.5
14		free/reduced	355	35.5
15	test_preparation_course	none	642	64.2
16		completed	358	35.8

In [29]:
```
# Burada kategorik değişkenler üzerinden bir istatistik çıkarılmıştır. Tabloda değişkenleri, değişkenlerin kategc
# kategorinin gözlem sayısı ve yüzdelik karşılığı görülmektedir.
```

7. Aykırı Gözlem Analizi

In [46]:
```
df_say = df.select_dtypes(include = ['float64', 'int64']) # sayısal değişkenler seçildi.
```

In [47]:
```
df_say.head() # sayısal değişkenlerin ilk 5 tanesi görüntülendi.
```

Out[47]:

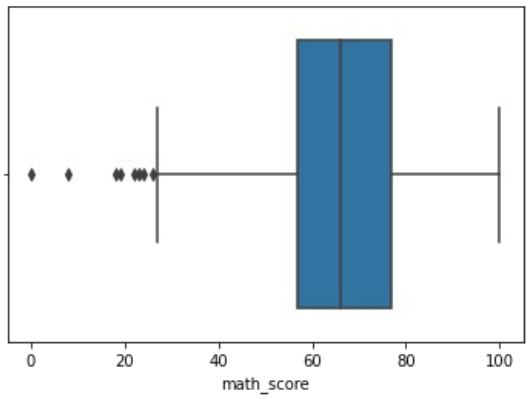
	math_score	reading_score	writing_score
0	72	72	74
1	69	90	88
2	90	95	93
3	47	57	44
4	76	78	75

math_score için aykırı gözlem analizi

In [48]:
```
df_math = df_say["math_score"]  # matematik dersi için ilk beş değer gözlemlendi.
df_math.head()
```

Out[48]:
```
0    72
1    69
2    90
3    47
4    76
Name: math_score, dtype: int64
```

In [49]:
```
sns.boxplot(x = df_math);
```



In [51]:
```
Q1 = df_math.quantile(0.25)
Q3 = df_math.quantile(0.75)
IQR = Q3-Q1
```

In [52]:
```
print(Q1)
```

```
print(Q3)
print(IQR)
```

```
57.0
77.0
20.0
```

In [53]:
```
alt_sinir = Q1- 1.5*IQR
ust_sinir = Q3 + 1.5*IQR
```

In [54]:
```
print(alt_sinir)
print(ust_sinir)
```

```
27.0
107.0
```

In [56]:
```
(df_math < alt_sinir) | (df_math > ust_sinir)
```

Out[56]:
```
0      False
1      False
2      False
3      False
4      False
       ...
995    False
996    False
997    False
998    False
999    False
Name: math_score, Length: 1000, dtype: bool
```

In [57]:
```
aykiri_tf = (df_math < alt_sinir) | (df_math > ust_sinir)
df_math[aykiri_tf] #aykırı değerler gözlemlendi.
```

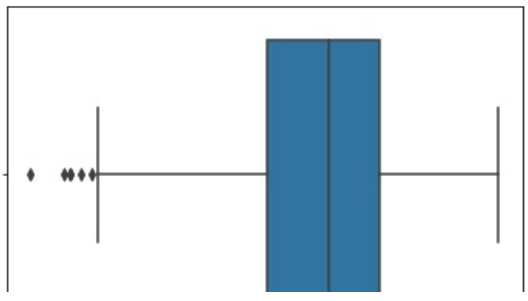
Out[57]:
```
17     18
59      0
145    22
338    24
466    26
787    19
842    23
980     8
Name: math_score, dtype: int64
```

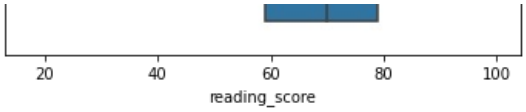
reading_score için aykırı gözlem analizi

In [64]:
```
df_reading = df_say["reading_score"]
df_reading.head()
```

Out[64]:
```
0    72
1    90
2    95
3    57
4    78
Name: reading_score, dtype: int64
```

In [66]:
```
sns.boxplot(x = df_reading);
```





In [67]:
```
Q1 = df_reading.quantile(0.25)
Q3 = df_reading.quantile(0.75)
IQR = Q3-Q1
```

In [68]:
```
print(Q1)
print(Q3)
print(IQR)
```

```
59.0
79.0
20.0
```

In [70]:
```
alt_sinir = Q1- 1.5*IQR
ust_sinir = Q3 + 1.5*IQR
```

In [71]:
```
print(alt_sinir)
print(ust_sinir)
```

```
29.0
109.0
```

In [73]:
```
(df_reading < alt_sinir) | (df_reading > ust_sinir)
```

Out[73]:
```
0      False
1      False
2      False
3      False
4      False
       ...
995    False
996    False
997    False
998    False
999    False
Name: reading_score, Length: 1000, dtype: bool
```

In [74]:
```
aykiri_tf = (df_reading < alt_sinir) | (df_reading > ust_sinir)
df_reading[aykiri_tf]
```

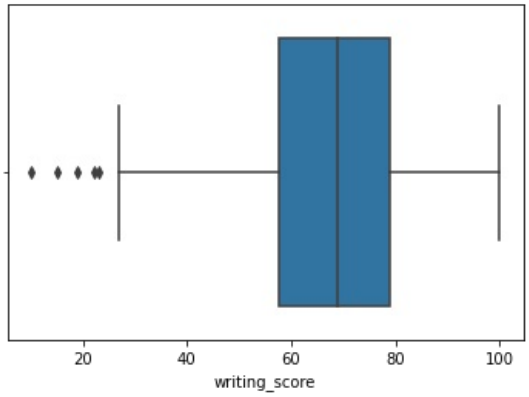
Out[74]:
```
59     17
76     26
211    28
327    23
596    24
980    24
Name: reading_score, dtype: int64
```

writing_score için aykırı gözlem analizi

In [75]:
```
df_writing = df_say["writing_score"]
df_writing.head()
```

Out[75]:
```
0    74
1    88
2    93
3    44
4    75
Name: writing_score, dtype: int64
```

In [76]:
```
sns.boxplot(x = df_writing);
```



In [77]:
```
Q1 = df_writing.quantile(0.25)
Q3 = df_writing.quantile(0.75)
IQR = Q3-Q1
```

In [78]:
```
print(Q1)
print(Q3)
print(IQR)
```

```
57.75
79.0
21.25
```

In [79]:
```
alt_sinir = Q1- 1.5*IQR
ust_sinir = Q3 + 1.5*IQR
```

In [80]:
```
print(alt_sinir)
print(ust_sinir)
```

```
25.875
110.875
```

In [82]:
```
(df_writing < alt_sinir) | (df_writing > ust_sinir)
```

Out[82]:
```
0      False
1      False
2      False
3      False
4      False
       ...
995    False
996    False
997    False
998    False
999    False
Name: writing_score, Length: 1000, dtype: bool
```

In [83]:
```
aykiri_tf = (df_writing < alt_sinir) | (df_writing > ust_sinir)
df_writing[aykiri_tf]
```

Out[83]:
```
59     10
76     22
327    19
596    15
980    23
Name: writing_score, dtype: int64
```

8.One-Hot Dönüşümü

One-hot dönüşümü, kategorik verileri makine öğrenimi modellerine uygun hale getirmek için kullanılan bir veri dönüşüm yöntemidir. Kategorik veriler, sayısal olmayan değerlerden oluşur ve doğrudan makine öğrenimi modellerine uygulanamazlar. Bu nedenle, kategorik verilerin sayısal formata dönüştürülmesi gerekmektedir.

In [67]:
```
df_one_hot = pd.get_dummies(df, columns = ["gender","ethnicity","parental_level_of_education","lunch","test_prepa
df_one_hot
```

Out[67]:

	math_score	reading_score	writing_score	gender_female	gender_male	ethnicity_group A	ethnicity_group B	ethnicity_group C	ethnicity_group D
0	72	72	74	1	0	0	1	0	0
1	69	90	88	1	0	0	0	1	0
2	90	95	93	1	0	0	1	0	0
3	47	57	44	0	1	1	0	0	0
4	76	78	75	0	1	0	0	1	0
...	...	...	...	...	...	...	...	...	...
995	88	99	95	1	0	0	0	0	0
996	62	55	55	0	1	0	0	1	0
997	59	71	65	1	0	0	0	1	0
998	68	78	77	1	0	0	0	0	1
999	77	86	86	1	0	0	0	0	1

1000 rows × 20 columns

9. Makine Öğrenmesi

In [37]:
```
pip install xgboost
```

```
Collecting xgboost
  Downloading xgboost-1.7.5-py3-none-win_amd64.whl (70.9 MB)
Requirement already satisfied: scipy in c:\anaconda3\lib\site-packages (from xgboost) (1.7.1)
Requirement already satisfied: numpy in c:\anaconda3\lib\site-packages (from xgboost) (1.20.3)
Installing collected packages: xgboost
Successfully installed xgboost-1.7.5
Note: you may need to restart the kernel to use updated packages.
```

In [39]:
```
pip install lightgbm
```

```
Collecting lightgbmNote: you may need to restart the kernel to use updated packages.

  Downloading lightgbm-3.3.5-py3-none-win_amd64.whl (1.0 MB)
Requirement already satisfied: scipy in c:\anaconda3\lib\site-packages (from lightgbm) (1.7.1)
Requirement already satisfied: scikit-learn!=0.22.0 in c:\anaconda3\lib\site-packages (from lightgbm) (0.24.2)
Requirement already satisfied: wheel in c:\anaconda3\lib\site-packages (from lightgbm) (0.37.0)
Requirement already satisfied: numpy in c:\anaconda3\lib\site-packages (from lightgbm) (1.20.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\anaconda3\lib\site-packages (from scikit-learn!=0.22.0-
>lightgbm) (2.2.0)
Requirement already satisfied: joblib>=0.11 in c:\anaconda3\lib\site-packages (from scikit-learn!=0.22.0->lightgb
m) (1.1.0)
Installing collected packages: lightgbm
Successfully installed lightgbm-3.3.5
```

In [41]:
```
pip install catboost
```

```
Collecting catboost
  Downloading catboost-1.2-cp39-cp39-win_amd64.whl (101.0 MB)
Requirement already satisfied: scipy in c:\anaconda3\lib\site-packages (from catboost) (1.7.1)
Collecting graphviz
  Downloading graphviz-0.20.1-py3-none-any.whl (47 kB)
Requirement already satisfied: six in c:\anaconda3\lib\site-packages (from catboost) (1.16.0)
Requirement already satisfied: matplotlib in c:\anaconda3\lib\site-packages (from catboost) (3.4.3)
Requirement already satisfied: numpy>=1.16.0 in c:\anaconda3\lib\site-packages (from catboost) (1.20.3)
Collecting plotly
  Downloading plotly-5.15.0-py2.py3-none-any.whl (15.5 MB)
Requirement already satisfied: pandas>=0.24 in c:\anaconda3\lib\site-packages (from catboost) (1.3.4)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\anaconda3\lib\site-packages (from pandas>=0.24->catbo
ost) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\anaconda3\lib\site-packages (from pandas>=0.24->catboost) (2021
.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\anaconda3\lib\site-packages (from matplotlib->catboost) (1
.3.1)
Requirement already satisfied: pillow>=6.2.0 in c:\anaconda3\lib\site-packages (from matplotlib->catboost) (8.4.0
```

```
)
Requirement already satisfied: cycler>=0.10 in c:\anaconda3\lib\site-packages (from matplotlib->catboost) (0.10.0
)
Requirement already satisfied: pyparsing>=2.2.1 in c:\anaconda3\lib\site-packages (from matplotlib->catboost) (3.
0.4)
Requirement already satisfied: packaging in c:\anaconda3\lib\site-packages (from plotly->catboost) (21.0)
Collecting tenacity>=6.2.0
  Downloading tenacity-8.2.2-py3-none-any.whl (24 kB)
Installing collected packages: tenacity, plotly, graphviz, catboost
Successfully installed catboost-1.2 graphviz-0.20.1 plotly-5.15.0 tenacity-8.2.2
Note: you may need to restart the kernel to use updated packages.
```

In [43]:
```
!pip install skompiler
```

```
Collecting skompiler
  Downloading SKompiler-0.7.tar.gz (45 kB)
Requirement already satisfied: scikit-learn>=0.22 in c:\anaconda3\lib\site-packages (from skompiler) (0.24.2)
Requirement already satisfied: joblib>=0.11 in c:\anaconda3\lib\site-packages (from scikit-learn>=0.22->skompiler
) (1.1.0)
Requirement already satisfied: scipy>=0.19.1 in c:\anaconda3\lib\site-packages (from scikit-learn>=0.22->skompile
r) (1.7.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\anaconda3\lib\site-packages (from scikit-learn>=0.22->s
kompiler) (2.2.0)
Requirement already satisfied: numpy>=1.13.3 in c:\anaconda3\lib\site-packages (from scikit-learn>=0.22->skompile
r) (1.20.3)
Building wheels for collected packages: skompiler
Failed to build skompiler
Installing collected packages: skompiler
    Running setup.py install for skompiler: started
    Running setup.py install for skompiler: finished with status 'done'
Successfully installed skompiler-0.7
```

```
  WARNING: Building wheel for skompiler failed: [WinError 5] Erişim engellendi: 'c:\\users\\i\u0307rem'
  DEPRECATION: skompiler was installed using the legacy 'setup.py install' method, because a wheel could not be b
uilt for it. A possible replacement is to fix the wheel build issue reported above. You can find discussion regar
ding this at https://github.com/pypa/pip/issues/8368.
```

In [74]:
```
import numpy as np
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
import seaborn as sns
from sklearn.preprocessing import scale
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn.metrics import roc_auc_score,roc_curve
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn import tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier


from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.tree import DecisionTreeClassifier
from skompiler import skompile



from warnings import filterwarnings
filterwarnings('ignore')
```

9.1. Doğrusal Regresyon

In [117…
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, cross_val_score, cross_val_predict
from sklearn.cross_decomposition import PLSRegression, PLSSVD
from sklearn.linear_model import Ridge
```

```
from sklearn.linear_model import Lasso
```

In [148…

```
y = df_one_hot["math_score"] # bağımlı değişken ataması
X = df_one_hot.drop(['math_score'], axis=1) # bağımsız değişkenlerin ataması
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.30,)
```

In [149…

```
models = {
    "Lineer Regression" : LinearRegression(),
    "PLS Regression" : PLSRegression(),
    "Ridge" : Ridge(alpha=0.1),
    "Lasso" : Lasso(alpha=0.1)
}
```

In [150…

```
for name,model in models.items():
    print(name, " :")
    lr=model.fit(X_train,y_train)
    y_pred = lr.predict(X_test)
    lr_pred_egitim = lr.predict(X_train)
    print('Training Score:',lr.score(X_train,y_train))
    print('Testing Score:',r2_score(y_test,y_pred))
    print('Diğer Matrikler: ')
    print('Test Hatası---> MSE:', mean_squared_error(y_test,y_pred))
    print('Eğitim Hatası ---> MSE',mean_squared_error(y_train,lr_pred_egitim))
   # print('MAE:',mean_absolute_error(y_test,y_pred))
    cross_val= cross_val_score(model, X_train, y_train, cv = 10, scoring="r2").mean()
    print("ort_r2:",cross_val)
    print(" ")
```

```
Lineer Regression  :
Training Score: 0.8856110885896209
Testing Score: 0.8516099558786383
Diğer Matrikler:
Test Hatası---> MSE: 31.23270120920146
Eğitim Hatası ---> MSE 27.194467527545736
ort_r2: 0.8714146057835522

PLS Regression  :
Training Score: 0.862139002331668
Testing Score: 0.8250191189804122
Diğer Matrikler:
Test Hatası---> MSE: 36.82946255975184
Eğitim Hatası ---> MSE 32.77464903006624
ort_r2: 0.8441108462409523

Ridge  :
Training Score: 0.8856110477905219
Testing Score: 0.8516185437635391
Diğer Matrikler:
Test Hatası---> MSE: 31.23089365637868
Eğitim Hatası ---> MSE 27.194477226997883
ort_r2: 0.8714205961328361

Lasso  :
Training Score: 0.8826203948929237
Testing Score: 0.851084832142182
Diğer Matrikler:
Test Hatası---> MSE: 31.34322771288782
Eğitim Hatası ---> MSE 27.905465836882634
ort_r2: 0.8705120222055036
```

9.2. Doğrusal olmayan Regresyon

In [123…

```
!pip install skompiler
!pip install astor
!pip install xgboost
!pip install catboost
!pip install lightgbm
```

```
Requirement already satisfied: skompiler in c:\anaconda3\lib\site-packages (0.7)
Requirement already satisfied: scikit-learn>=0.22 in c:\anaconda3\lib\site-packages (from skompiler) (0.24.2)
Requirement already satisfied: scipy>=0.19.1 in c:\anaconda3\lib\site-packages (from scikit-learn>=0.22->skompile
r) (1.7.1)
Requirement already satisfied: numpy>=1.13.3 in c:\anaconda3\lib\site-packages (from scikit-learn>=0.22->skompile
r) (1.20.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\anaconda3\lib\site-packages (from scikit-learn>=0.22->s
kompiler) (2.2.0)
Requirement already satisfied: joblib>=0.11 in c:\anaconda3\lib\site-packages (from scikit-learn>=0.22->skompiler
) (1.1.0)
```

```
Collecting astor
  Downloading astor-0.8.1-py2.py3-none-any.whl (27 kB)
Installing collected packages: astor
Successfully installed astor-0.8.1
Requirement already satisfied: xgboost in c:\anaconda3\lib\site-packages (1.7.5)
Requirement already satisfied: numpy in c:\anaconda3\lib\site-packages (from xgboost) (1.20.3)
Requirement already satisfied: scipy in c:\anaconda3\lib\site-packages (from xgboost) (1.7.1)
Requirement already satisfied: catboost in c:\anaconda3\lib\site-packages (1.2)
Requirement already satisfied: scipy in c:\anaconda3\lib\site-packages (from catboost) (1.7.1)
Requirement already satisfied: six in c:\anaconda3\lib\site-packages (from catboost) (1.16.0)
Requirement already satisfied: pandas>=0.24 in c:\anaconda3\lib\site-packages (from catboost) (1.3.4)
Requirement already satisfied: numpy>=1.16.0 in c:\anaconda3\lib\site-packages (from catboost) (1.20.3)
Requirement already satisfied: matplotlib in c:\anaconda3\lib\site-packages (from catboost) (3.4.3)
Requirement already satisfied: graphviz in c:\anaconda3\lib\site-packages (from catboost) (0.20.1)
Requirement already satisfied: plotly in c:\anaconda3\lib\site-packages (from catboost) (5.15.0)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\anaconda3\lib\site-packages (from pandas>=0.24->catbo
ost) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\anaconda3\lib\site-packages (from pandas>=0.24->catboost) (2021
.3)
Requirement already satisfied: pillow>=6.2.0 in c:\anaconda3\lib\site-packages (from matplotlib->catboost) (8.4.0
)
Requirement already satisfied: pyparsing>=2.2.1 in c:\anaconda3\lib\site-packages (from matplotlib->catboost) (3.
0.4)
Requirement already satisfied: cycler>=0.10 in c:\anaconda3\lib\site-packages (from matplotlib->catboost) (0.10.0
)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\anaconda3\lib\site-packages (from matplotlib->catboost) (1
.3.1)
Requirement already satisfied: tenacity>=6.2.0 in c:\anaconda3\lib\site-packages (from plotly->catboost) (8.2.2)
Requirement already satisfied: packaging in c:\anaconda3\lib\site-packages (from plotly->catboost) (21.0)
Requirement already satisfied: lightgbm in c:\anaconda3\lib\site-packages (3.3.5)
Requirement already satisfied: wheel in c:\anaconda3\lib\site-packages (from lightgbm) (0.37.0)
Requirement already satisfied: scikit-learn!=0.22.0 in c:\anaconda3\lib\site-packages (from lightgbm) (0.24.2)
Requirement already satisfied: numpy in c:\anaconda3\lib\site-packages (from lightgbm) (1.20.3)
Requirement already satisfied: scipy in c:\anaconda3\lib\site-packages (from lightgbm) (1.7.1)
Requirement already satisfied: joblib>=0.11 in c:\anaconda3\lib\site-packages (from scikit-learn!=0.22.0->lightgb
m) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\anaconda3\lib\site-packages (from scikit-learn!=0.22.0-
>lightgbm) (2.2.0)
```

In [124…

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV,cross_val_score
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.svm import SVR
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
from sklearn.preprocessing import scale
from sklearn import model_selection
from sklearn.tree import DecisionTreeRegressor, DecisionTreeClassifier
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import BaggingRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.svm import LinearSVR
from sklearn.preprocessing import StandardScaler


from skompiler import skompile
import xgboost as xgb
from xgboost import XGBRegressor
from catboost import CatBoostRegressor
from lightgbm import LGBMRegressor
from lightgbm import LGBMRegressor

from warnings import filterwarnings
filterwarnings('ignore')

from datetime import datetime
```

In [151…

```
y = df_one_hot["math_score"] # bağımlı değişken ataması
X = df_one_hot.drop(['math_score'], axis=1) # bağımsız değişkenlerin ataması
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.30,)
```

In [152…

```
models = {
    "KNN" : KNeighborsRegressor(),
    "LinearSVR" : LinearSVR(),
    "Yapay Sinir Ağları" : MLPRegressor(hidden_layer_sizes = (100,20)),
    "Karar Ağaçları" : DecisionTreeRegressor(),
    "Bagged Trees Regresyon" : BaggingRegressor(bootstrap_features = True),
    "Random Forests" : RandomForestRegressor(random_state = 42),
    "Gradient Boosting Machines " : GradientBoostingRegressor(),
```

```
    "XGBoost": XGBRegressor(),
    "Light GBM": LGBMRegressor(),


}
```

In [153…

```
for name,model in models.items():
    print(name, " :")
    lr=model.fit(X_train,y_train)
    y_pred = lr.predict(X_test)
    lr_pred_egitim = lr.predict(X_train)
    print('Training Score:',lr.score(X_train,y_train))
    print('Testing Score:',r2_score(y_test,y_pred))
    print('Diğer Matrikler: ')
    print('Test Hatası---> MSE:', mean_squared_error(y_test,y_pred))
    print('Eğitim Hatası ---> MSE',mean_squared_error(y_train,lr_pred_egitim))
   # print('MAE:',mean_absolute_error(y_test,y_pred))
    cross_val= cross_val_score(model, X_train, y_train, cv = 10, scoring="r2").mean()
    print("ort_r2:",cross_val)
    print(" ")
```

```
KNN  :
Training Score: 0.816644955866481
Testing Score: 0.726576630672336
Diğer Matrikler:
Test Hatası---> MSE: 68.7332
Eğitim Hatası ---> MSE 40.39514285714286
ort_r2: 0.7084451460907986

LinearSVR  :
Training Score: 0.8754036517214765
Testing Score: 0.8612670173730379
Diğer Matrikler:
Test Hatası---> MSE: 34.87471412901918
Eğitim Hatası ---> MSE 27.449952696824585
ort_r2: 0.862347351517055

Yapay Sinir Ağları  :
Training Score: 0.8772960984506015
Testing Score: 0.8601980989242372
Diğer Matrikler:
Test Hatası---> MSE: 35.14341897932429
Eğitim Hatası ---> MSE 27.033025765068782
ort_r2: 0.8612914283534003

Karar Ağaçları  :
Training Score: 0.999905977092722
Testing Score: 0.6991276292463857
Diğer Matrikler:
Test Hatası---> MSE: 75.63333333333334
Eğitim Hatası ---> MSE 0.020714285714285713
ort_r2: 0.7410044550604976

Bagged Trees Regresyon  :
Training Score: 0.9004812794771813
Testing Score: 0.7501406159528095
Diğer Matrikler:
Test Hatası---> MSE: 62.809682500151915
Eğitim Hatası ---> MSE 21.925074117687885
ort_r2: 0.7737921411139259

Random Forests  :
Training Score: 0.9802898426664244
Testing Score: 0.8131060519049251
Diğer Matrikler:
Test Hatası---> MSE: 46.981423514733564
Eğitim Hatası ---> MSE 4.342365518162456
ort_r2: 0.8495501973072284

Gradient Boosting Machines   :
Training Score: 0.9176931797691324
Testing Score: 0.8421686636401238
Diğer Matrikler:
Test Hatası---> MSE: 39.67566062464234
Eğitim Hatası ---> MSE 18.133102239183327
ort_r2: 0.8648528093193001

XGBoost  :
Training Score: 0.9977528459857788
Testing Score: 0.7980333017366904
Diğer Matrikler:
Test Hatası---> MSE: 50.770413294249444
Eğitim Hatası ---> MSE 0.4950728672638083
ort_r2: 0.8158793206480887

Light GBM  :
Training Score: 0.945633597033531
```

```
Testing Score: 0.8205105710106535
Diğer Matrikler:
Test Hatası---> MSE: 45.12007459693879
Eğitim Hatası ---> MSE 11.977519488693664
ort_r2: 0.8493926889934713
```

9.3. Sınıflandırma Problemleri

In [132…

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
import seaborn as sns
from sklearn.preprocessing import scale
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn.metrics import roc_auc_score,roc_curve
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn import tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier


from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.tree import DecisionTreeClassifier
from skompiler import skompile



from warnings import filterwarnings
filterwarnings('ignore')
```

In [155…

```
y = df_one_hot["gender_female"] # bağımlı değişken ataması
X = df_one_hot.drop(['gender_female'], axis=1) # bağımsız değişkenlerin ataması
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.30,)
```

In [156…

```
models = {
    "Lojistik Regresyon" : LogisticRegression(solver = "liblinear"),
    "Gaussian Naive Bayes" : GaussianNB(),
    "KNN" : KNeighborsClassifier(),
    "SVC" : SVC(kernel = "linear"),
    "RBF SVC" : SVC(kernel = "rbf"),
    "Karar Ağaçları" : DecisionTreeClassifier(),
    "Random Forests" : RandomForestClassifier(),
    "Gradient Boosting Machines " : GradientBoostingClassifier(),
    "XGBoost": XGBClassifier(),
    "Light GBM": LGBMClassifier(),


}
```

In [157…

```
from sklearn.model_selection import cross_val_score

for model_name, model in models.items():
    scores = cross_val_score(model, X_train, y_train, cv=5)
    mean_score = scores.mean()
    print(f"{model_name} accuracy: {mean_score}")
```

```
Lojistik Regresyon accuracy: 1.0
Gaussian Naive Bayes accuracy: 1.0
KNN accuracy: 0.8657142857142857
SVC accuracy: 1.0
RBF SVC accuracy: 0.8757142857142857
Karar Ağaçları accuracy: 1.0
Random Forests accuracy: 1.0
Gradient Boosting Machines  accuracy: 1.0
XGBoost accuracy: 1.0
Light GBM accuracy: 1.0
```

In [ ]: